RowArmor: Efficient and Comprehensive Protection Against DRAM Disturbance Errors

Minbok Wi†, Yoonyul Yoo‡, Yoojin Kim‡, Jaeho Shin‡, Jumin Kim†, Yesin Ryu‡,
Saeid Gorgin‡, Jung Ho Ahn†, and Jungrae Kim‡
†Seoul National University, ‡Sungkyunkwan University
Presenter: Jumin Kim (jumin.kim@scale.snu.ac.kr)

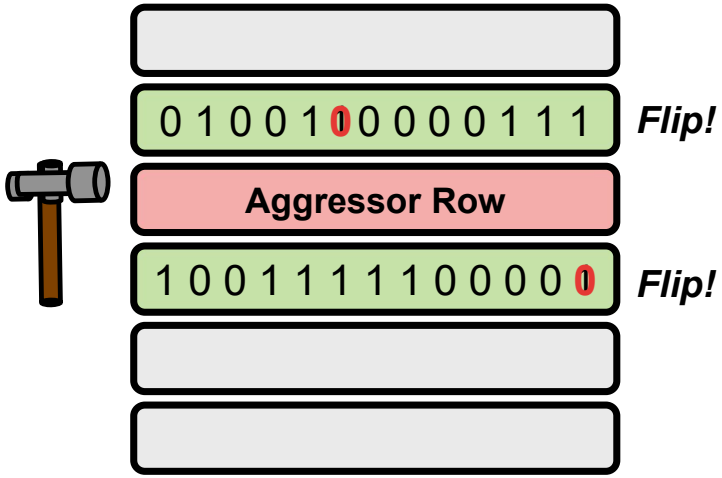
Overview

- RowHammer: trend and challenges
- Limitations of existing preventive defenses
- Reactive defenses for RowHammer
- RowArmor
  - Error confinement
  - Error correction
  - Row address obfuscation
  - Guardband scrubbing
- Evaluation
- Summary

RowHammer

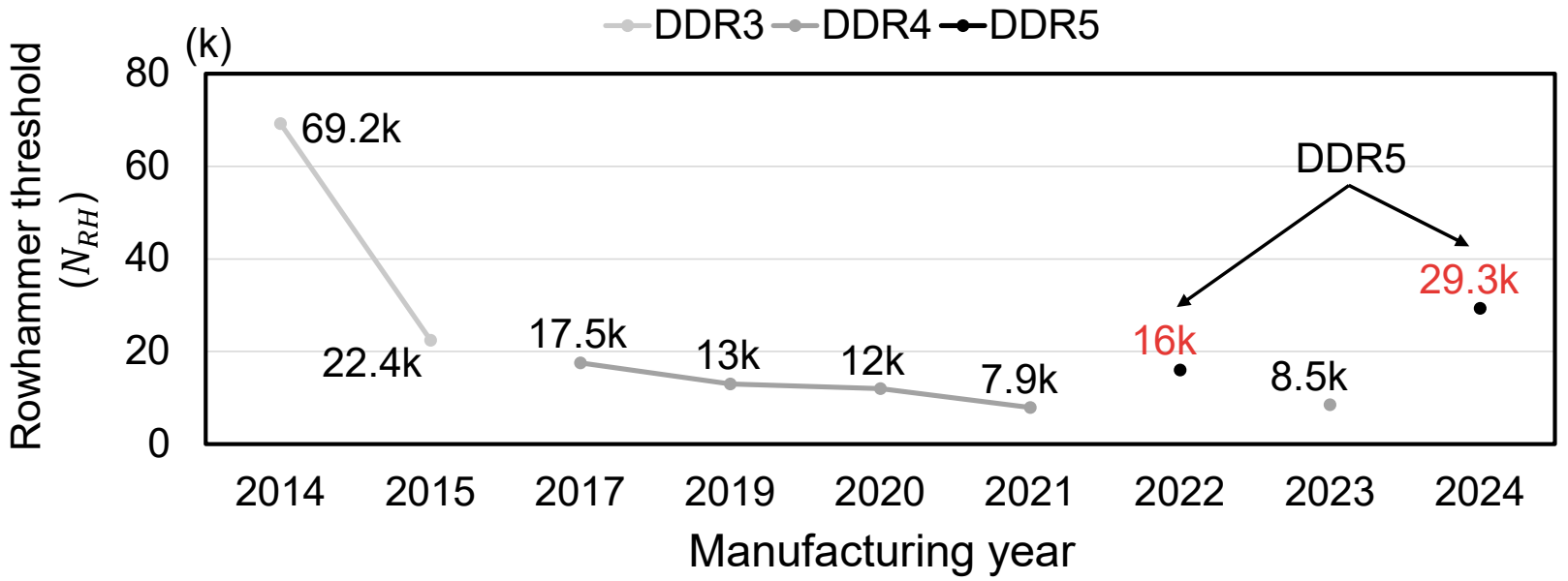
RowHammer (RH)

- Repeatedly accessing (hammering) a specific (aggressor) row results in bitflips in its adjacent (victim) rows.
- When the number of hammering exceeds a specific threshold (N_{RH}), the bitflips occur.



RowHammer vulnerability trend

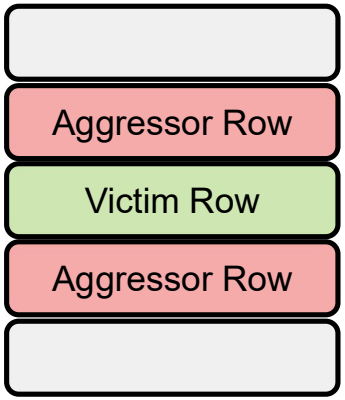
- RowHammer threshold (N_{RH}) continues to decrease with DRAM scaling[1-6].
  - DDR5 is still vulnerable, even with an on-die ECC (OECC) enabled.



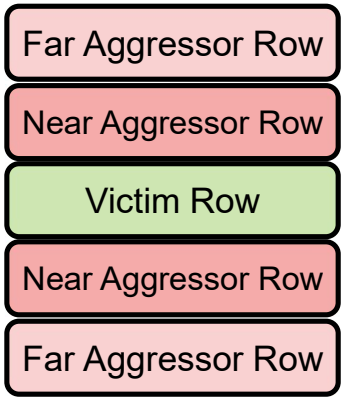
[1] Y. Kim et al., “Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors,” ISCA, 2014.
[2] J. S. Kim *et al.*, "Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques," *ISCA*, 2020.
[3] H. Hassan *et al.*, “Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom Row Hammer Patterns, and Implications," MICRO, 2021.
[4] H. Luo *et al.*, "An Experimental Characterization of Combined RowHammer and RowPress Read Disturbance in Modern DRAM Chips,” DSN-S, 2024.
[5] S. Gloor *et al.*, "REFault: A Fault Injection Platform for Rowhammer Research on DDR5 Memory," *Microarchitecture Security Conference*, 2025.
[6] D. Meyer *et al.*, “Phoenix: Rowhammer Attacks on DDR5 with Self-Correcting Synchronization,” IEEE S&P, 2026.

RowHammer threshold (N_{RH})

- Targeted attacks
  - Single-sided, Double-sided, Many-sided attack
- More sophisticated attack patterns
  - Half-Double [1] disturbs non-adjacent DRAM rows.



Double-sided attack

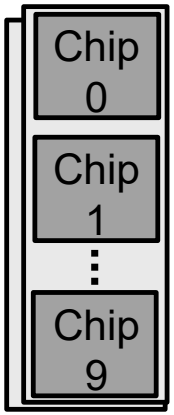


Half-Double
[1]

[1] Kogler, Andreas, et al. "Half-Double: Hammering from the next row over." *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

RowHammer threshold ($\mathbf{N_{RH}}$)

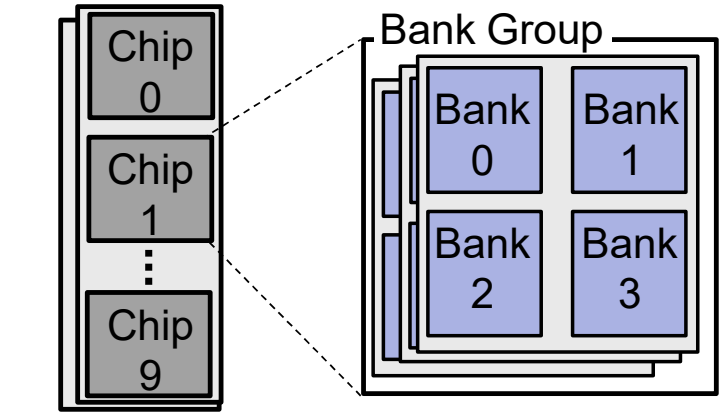
- Highly variable vulnerability
  - $\mathbf{N_{RH}}$ varies widely.



Chip variation

RowHammer threshold (N_{RH})

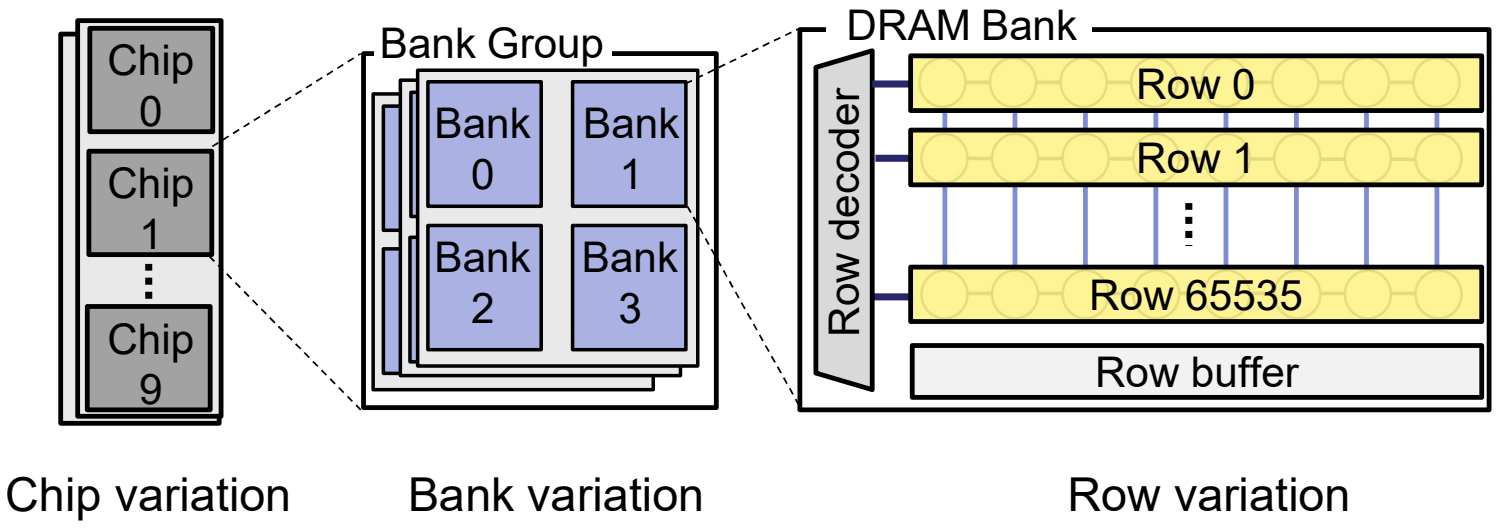
- Highly variable vulnerability
  - N_{RH} varies widely.



Chip variation Bank variation

RowHammer threshold (N_{RH})

- Highly variable vulnerability
  - N_{RH} varies widely.
  - Up to **16x** variation[1] across rows, banks, chips.

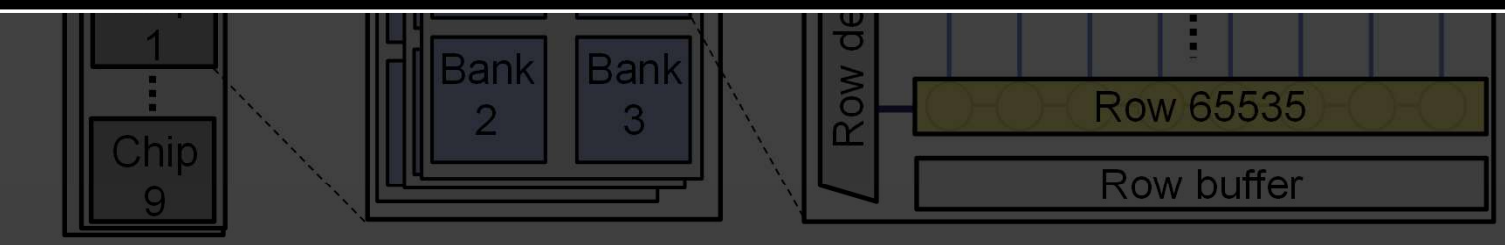


[1] Yağlıkçı, Abdullah Giray, et al. "Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions." *HPCA*. IEEE, 2024.

RowHammer threshold (N_{RH})

- Highly variable vulnerability
  - N_{RH} varies widely.
  - Up to **16x** variation[1] across rows, banks, chips.

1. The N_{RH} depends on the attack patterns.
2. The N_{RH} exhibits significant spatial variation.



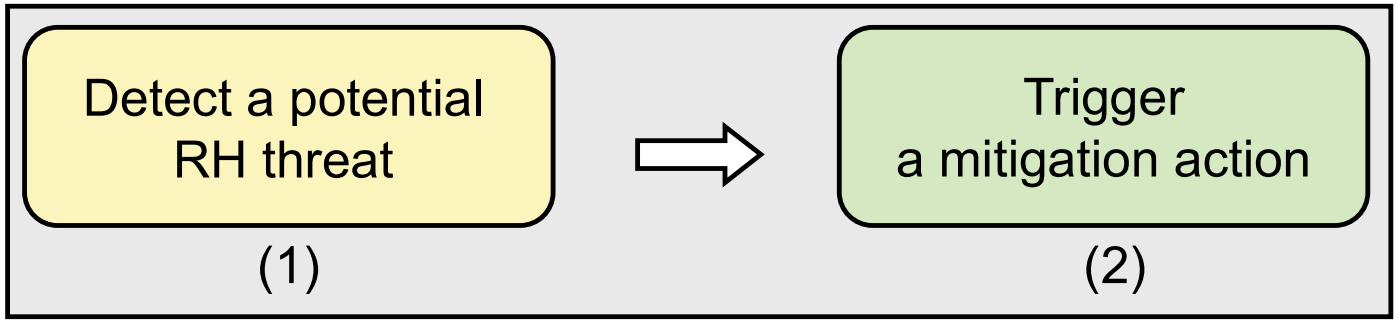
Chip variation Bank variation Row variation

[1] Yağlıkçı, Abdullah Giray, et al. "Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions." *HPCA*. IEEE, 2024.

Existing RowHammer Mitigations

Preventive mitigations

- Most existing RH mitigations are preventive.
  - Preventive mitigations attempt to stop attacks **before bitflips occur**.
  - (1) Detect potential RH threat and (2) Trigger mitigation action



Preventive mitigations

Preventive mitigations

Preventive Mitigation Scheme	Deterministic	Detection	Mitigation	Implemented place	Storage overhead
PARA	Probabilistic	None	Victim refresh	MC	None
Graphene	Deterministic	Count-Based	Victim refresh	MC	High (SRAM)
BlockHammer	Deterministic	Bloom-Filters	Agg. throttle	MC	High (Filter)
Hydra	Deterministic	Hybrid Count-Based	Victim refresh / Delays	MC	Mid (SRAM / DRAM)
SRS	Probabilistic	Count-Based	Agg. swap	MC	Mid (SRAM)
ABACuS	Deterministic	Count-Based	Victim refresh	MC	Low (SRAM)
START	Deterministic	Count-Based (in cache)	Victim refresh	MC & LLC	Low (MC+LLC)
Rubix	Probabilistic	None	Line-to-row remapping	MC	Low
PRAC	Deterministic	Count-Based (in DRAM)	Victim refresh	MC & DRAM	Low
MOAT	Deterministic	Count-Based (in DRAM)	Victim refresh	MC & DRAM	Low
QPRAC	Deterministic	Count-Based (in DRAM)	Victim refresh	MC & DRAM	Low

Limitations of preventive mitigations

- **Pessimistic** threshold
  - Large $\mathrm{N_{RH}}$ variation across attack patterns and spatial locations.
  - Mitigation thresholds must be set for the **worst-case N_{RH}**, triggering unnecessary mitigation actions.

Limitations of preventive mitigations

- **Pessimistic** threshold
  - Large $\mathrm{N_{RH}}$ variation across attack patterns and spatial locations.
  - Mitigation thresholds must be set for the **worst-case N_{RH}**, triggering unnecessary mitigation actions.
- **Performance overhead** due to **frequent** mitigation
  - Refresh-based defenses trigger excessive refresh operations [1-2].
  - PRAC can consume up to 94% of memory bandwidth [3].
  - Can be exploited for Denial-of-Service (DoS) attacks.

[1] Canpolat, Oğuzhan, et al. "Breakhammer: Enhancing rowhammer mitigations by carefully throttling suspect threads." *MICRO*. IEEE, 2024.
[2] Nazaraliyev, Ravan, et al. "Not so Refreshing: Attacking GPUs using RFM Rowhammer Mitigation." in *USENIX Security*, 2025.
[3] O. Canpolat et al., “Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance,” in *HPCA*, 2025.

Limitations of preventive mitigations

- **Pessimistic** threshold
  - Large $\mathrm{N_{RH}}$ variation across attack patterns and spatial locations.
  - Mitigation thresholds must be set for the **worst-case N_{RH}**, triggering unnecessary mitigation actions.
- **Worst-case thresholds trigger frequent mitigation, degrading performance.**
  - Can be exploited for Denial-of-Service (DoS) attacks.

[1] Canpolat, Oğuzhan, et al. "Breakhammer: Enhancing rowhammer mitigations by carefully throttling suspect threads." *MICRO*. IEEE, 2024.
[2] Nazaraliyev, Ravan, et al. "Not so Refreshing: Attacking GPUs using RFM Rowhammer Mitigation." in *USENIX Security*, 2025.
[3] O. Canpolat et al., “Chronus: Understanding and Securing the Cutting-Edge Industry Solutions to DRAM Read Disturbance,” in *HPCA*, 2025.

Reactive RowHammer Mitigations

Reactive RH mitigations

- Allow disturbance errors but handle them safely.
  - Detect and correct errors before they affect the system.

Reactive RH mitigations

- Allow disturbance errors but handle them safely.
  - Detect and correct errors before they affect the system.
- Leverage existing DRAM reliability features, such as Error Correcting Code (**ECC)**.

Error Correcting Code (ECC)

- Modern systems already invest heavily in ECC for reliability.

Error Correcting Code (ECC)

- Modern systems already invest heavily in ECC for reliability.
- On-die ECC (OECC): Single-bit Error Correction (**SEC**)
  - (136,128) bit-level Codeword
  - Additional 6.25% redundancy

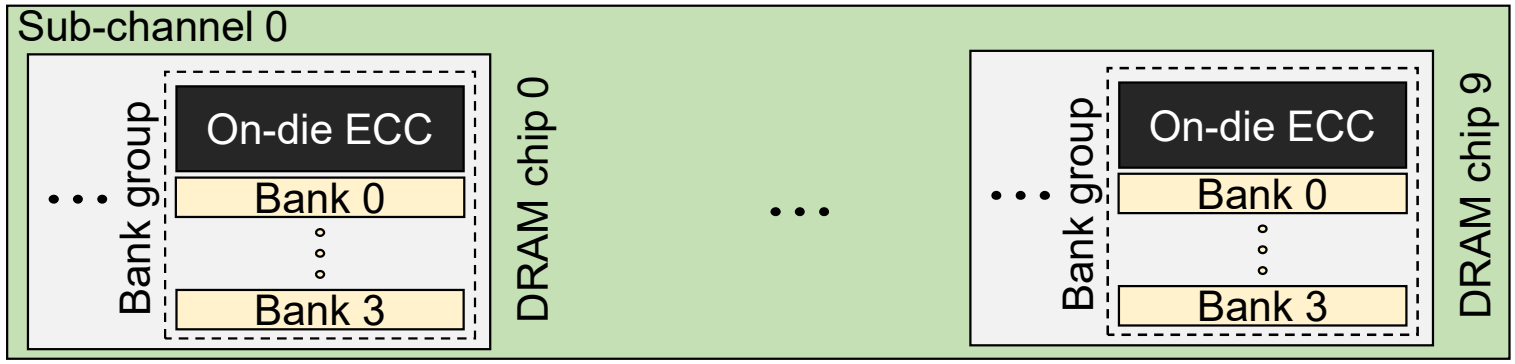
Error Correcting Code (ECC)

- Modern systems already invest heavily in ECC for reliability.
- On-die ECC (OECC): Single-bit Error Correction (**SEC**)
  - (136,128) bit-level Codeword
  - Additional 6.25% redundancy
- Memory-Controller (MC) ECC: Single-symbol Error Correction (**Chipkill** [1])
  - (10,8) symbol-level Codeword for DDR5 x4 chip.
  - Additional 25% redundancy

[1] Advanced Micro Devices (AMD), Inc. 2013. BIOS and Kernel Developer's Guide (BKDG) for AMD Family 15h Models 00h–0Fh Processors (rev. 3.24 ed.). Advanced Micro Devices (AMD)

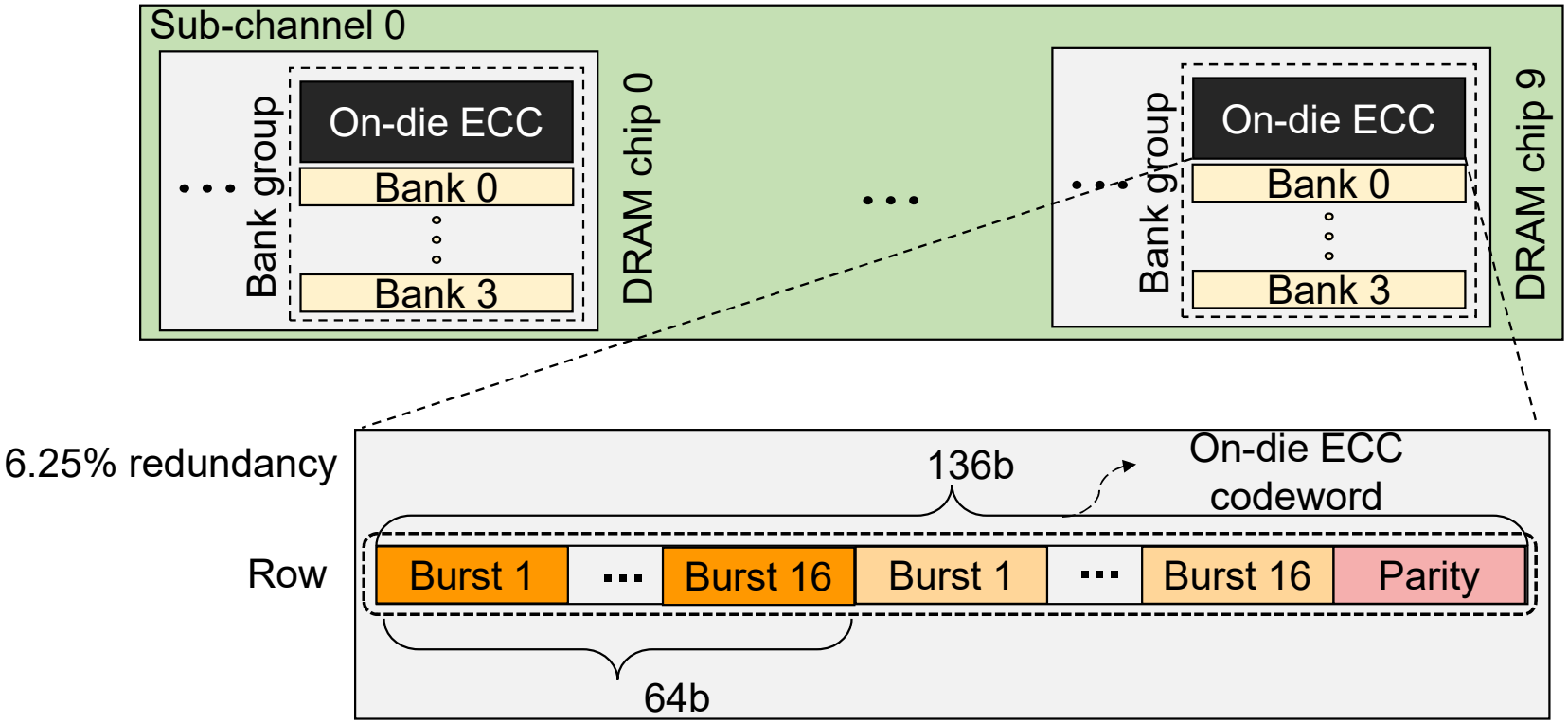
On-die ECC

- On-die ECC (OECC): Single error correction
  - **DDR5 ×4** chips with a burst length of 16



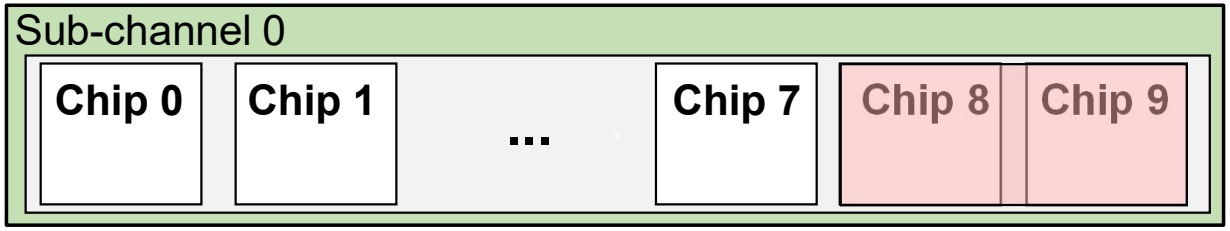
On-die ECC

- On-die ECC (OECC): Single error correction
  - **DDR5 ×4** chips with a burst length of 16



Chipkill ECC

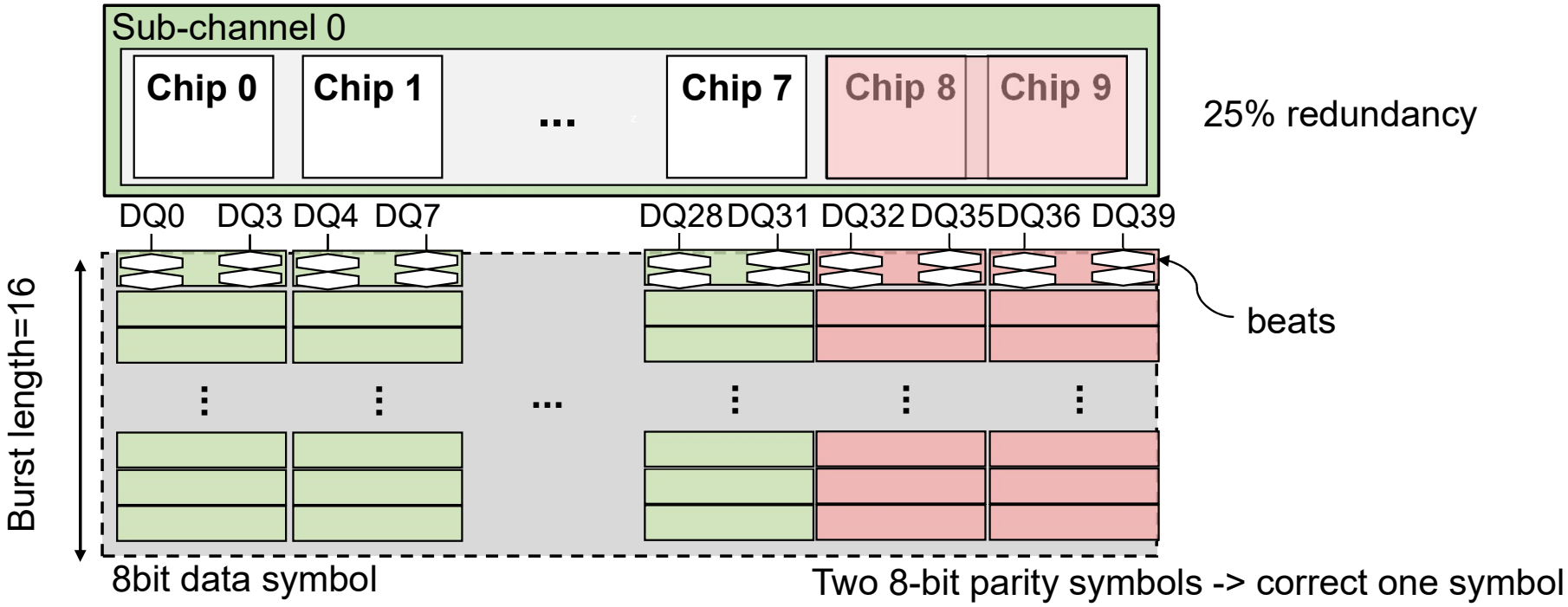
- 8-bit Symbol Chipkill
  - **DDR5 ×4** chips with a burst length of 16



25% redundancy

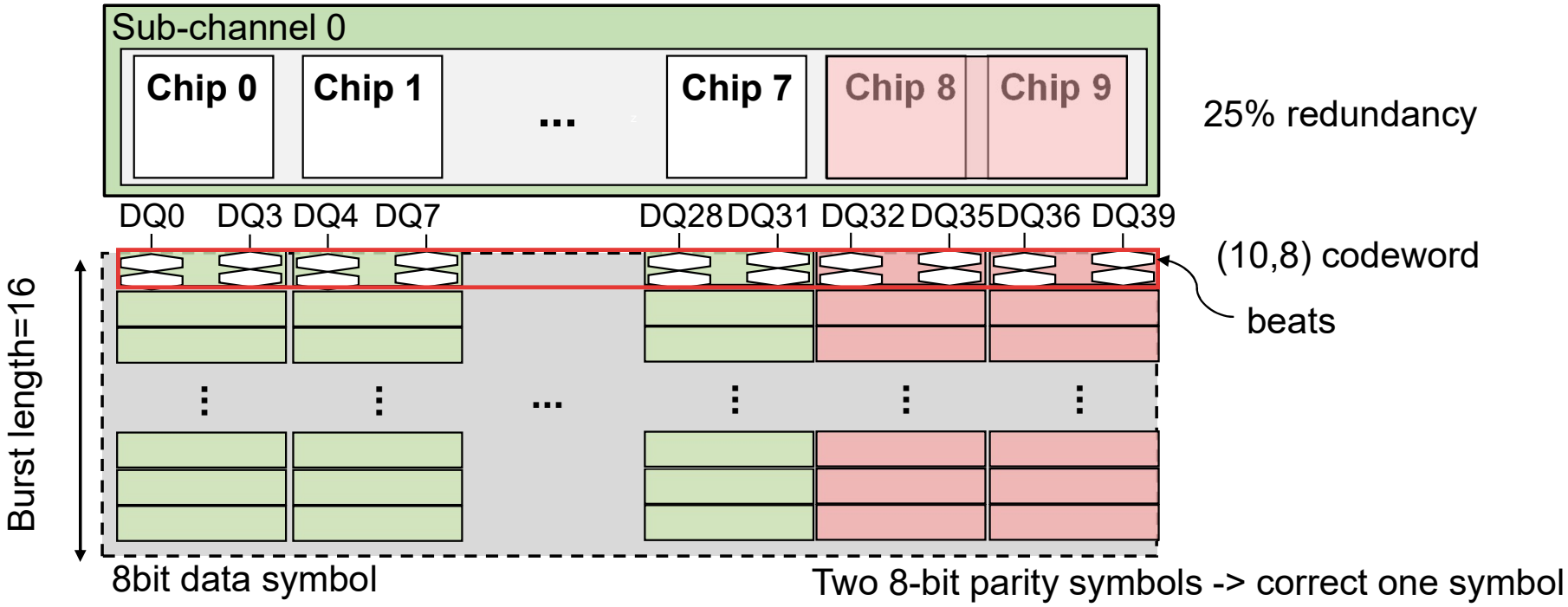
Chipkill ECC

- 8-bit Symbol Chipkill
  - **DDR5 ×4** chips with a burst length of 16



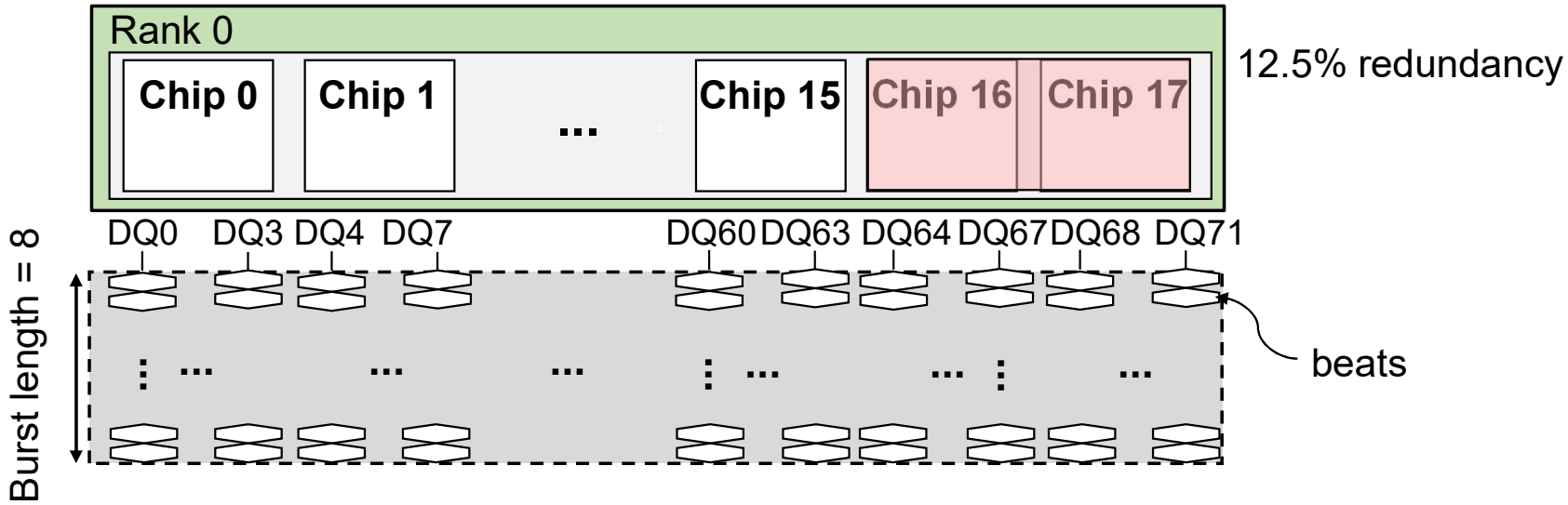
Chipkill ECC

- 8-bit Symbol Chipkill
  - **DDR5 ×4** chips with a burst length of 16



Bamboo ECC [1]

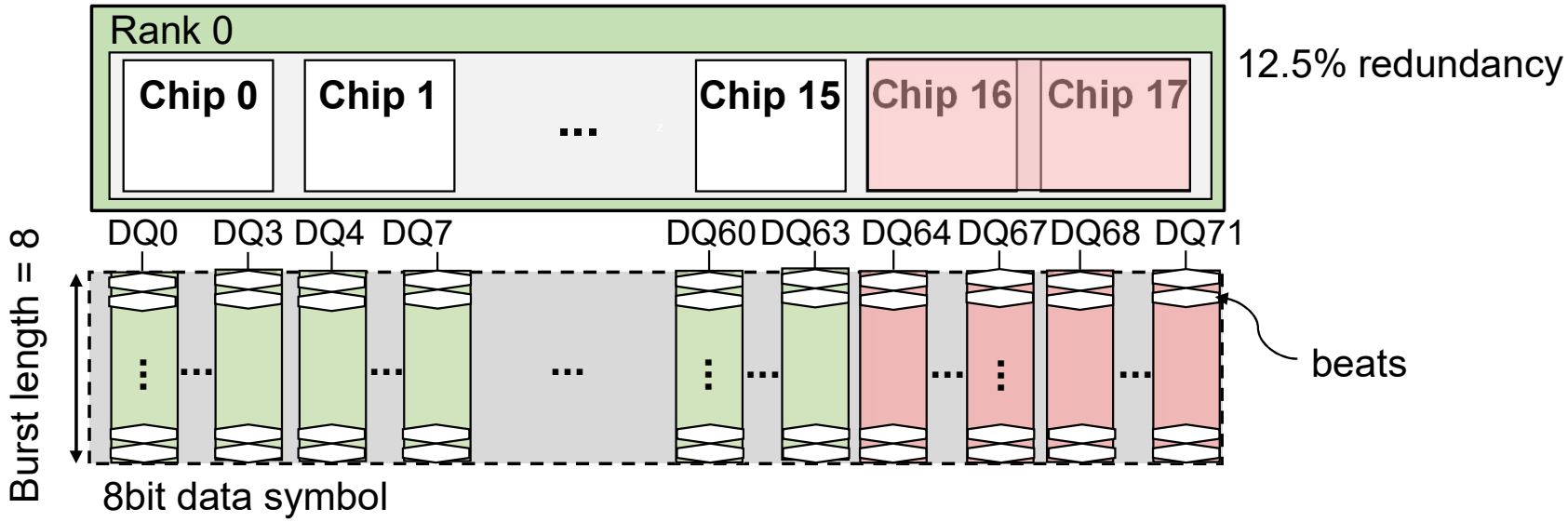
- Quadruple Pin Correcting (QPC)
  - 8-bit Symbol Chipkill
  - **DDR4 ×4** chips with a burst length of 8



[1] Jungrae Kim, Michael B. Sullivan, and Mattan Erez. “Bamboo ECC: Strong, Safe, and Flexible Codes for Reliable Computer Memory”. In IEEE HPCA. 2015

Bamboo ECC [1]

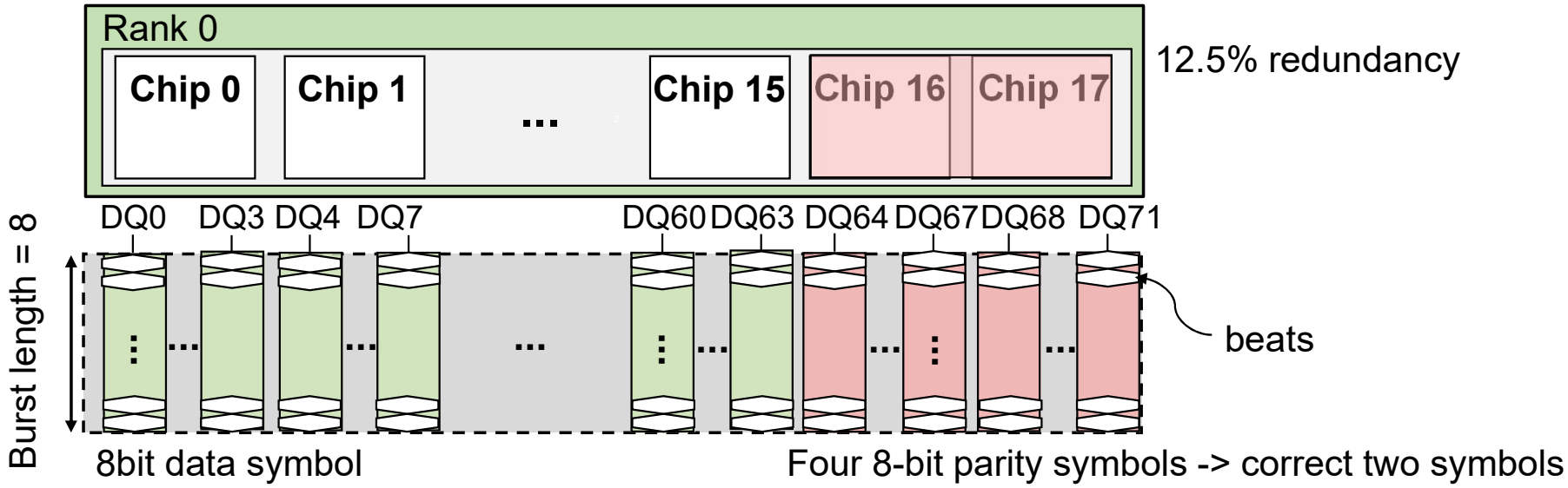
- Quadruple Pin Correcting (QPC)
  - 8-bit Symbol Chipkill
  - **DDR4 ×4** chips with a burst length of 8



[1] Jungrae Kim, Michael B. Sullivan, and Mattan Erez. “Bamboo ECC: Strong, Safe, and Flexible Codes for Reliable Computer Memory”. In IEEE HPCA. 2015

Bamboo ECC [1]

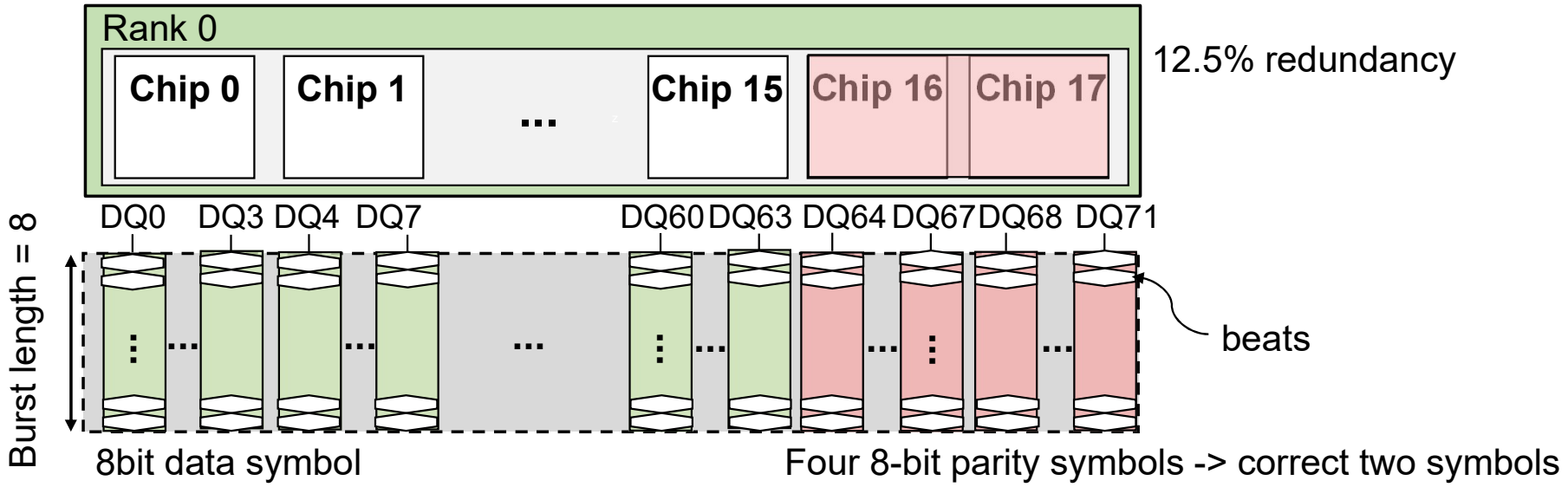
- Corrects up to **four independent pin errors** (stronger than Chipkill).



[1] Jungrae Kim, Michael B. Sullivan, and Mattan Erez. “Bamboo ECC: Strong, Safe, and Flexible Codes for Reliable Computer Memory”. In IEEE HPCA. 2015

Bamboo ECC [1]

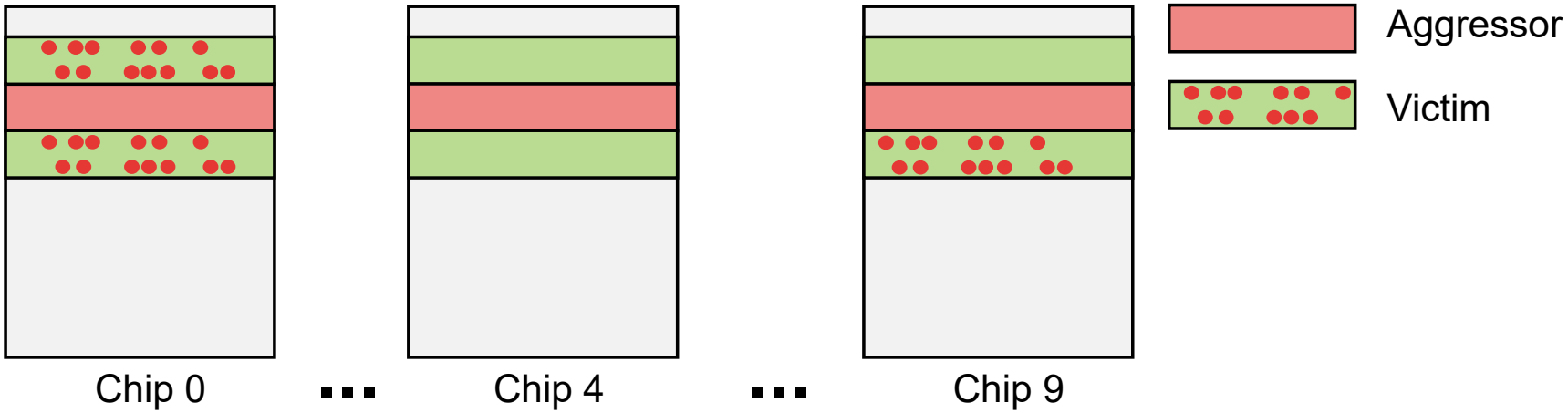
- Corrects up to **four independent pin errors** (stronger than Chipkill).
- Requires the full burst to form a codeword, increasing read latency by four cycles.



[1] Jungrae Kim, Michael B. Sullivan, and Mattan Erez. “Bamboo ECC: Strong, Safe, and Flexible Codes for Reliable Computer Memory”. In IEEE HPCA. 2015

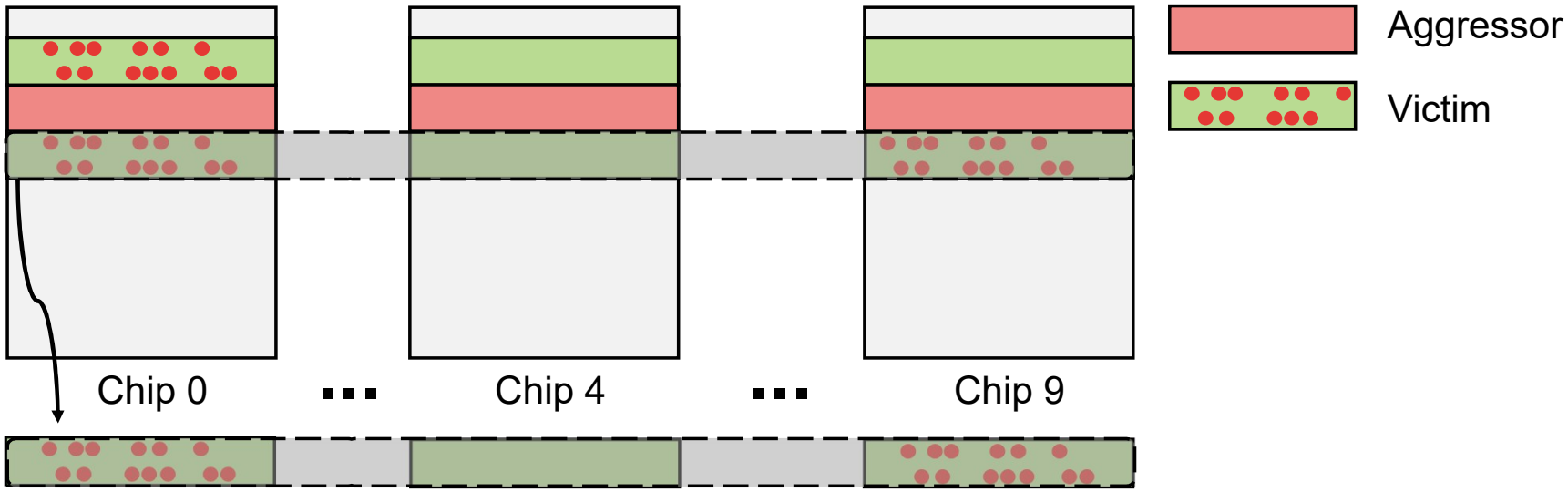
Limitation of conventional ECC for RH (1)

- 1) RH attacks can **accumulate** and overwhelm conventional ECC.



Limitation of conventional ECC for RH (1)

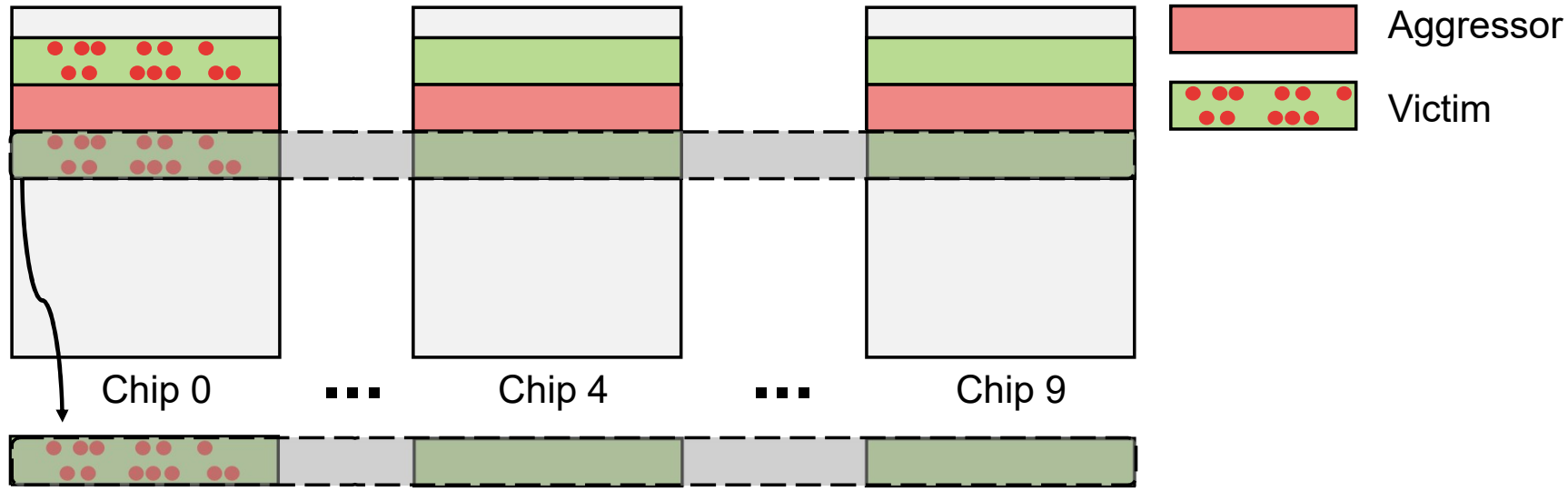
- 1) RH attacks can **accumulate** and overwhelm conventional ECC.
  - Victims are concentrated to two Chipkill codewords.



2 symbol error – uncorrectable by Chipkill

Limitation of conventional ECC for RH (2)

- 2) ECC corrects errors on memory access. Corrected values are **not written back** to DRAM.

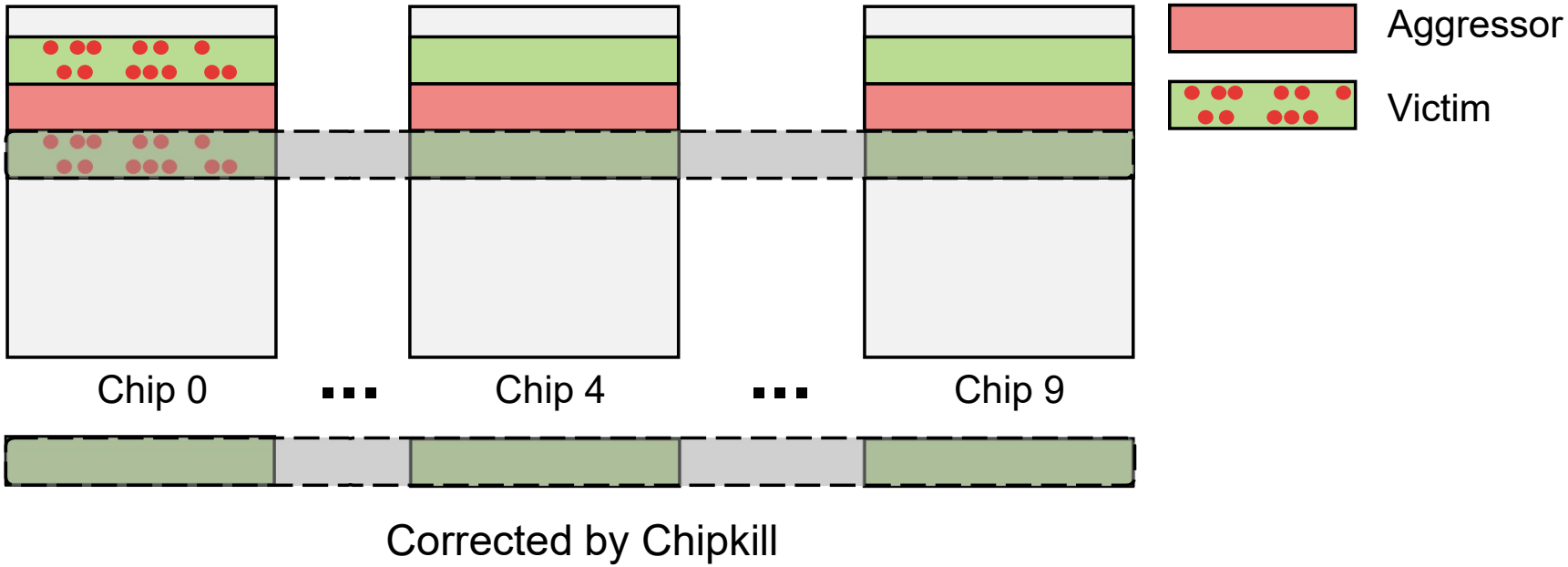


1 symbol error – correctable by Chipkill

[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Limitation of conventional ECC for RH (2)

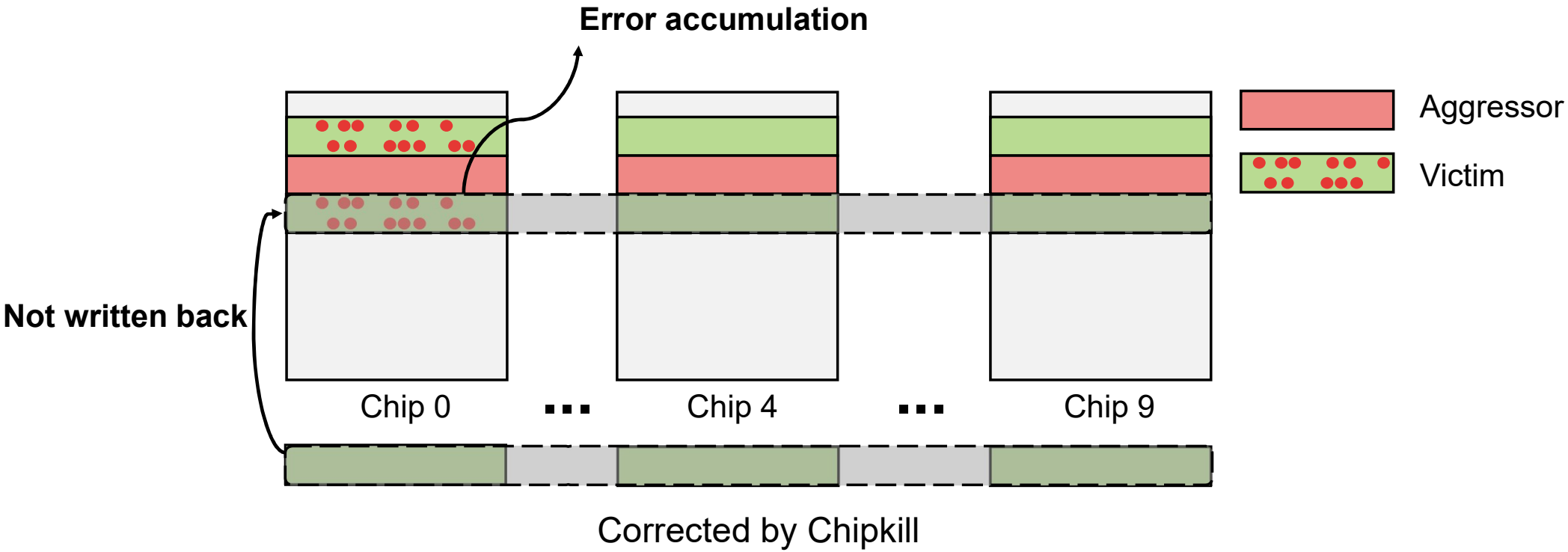
- 2) ECC corrects errors on memory access. Corrected values are **not written back** to DRAM.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Limitation of conventional ECC for RH (2)

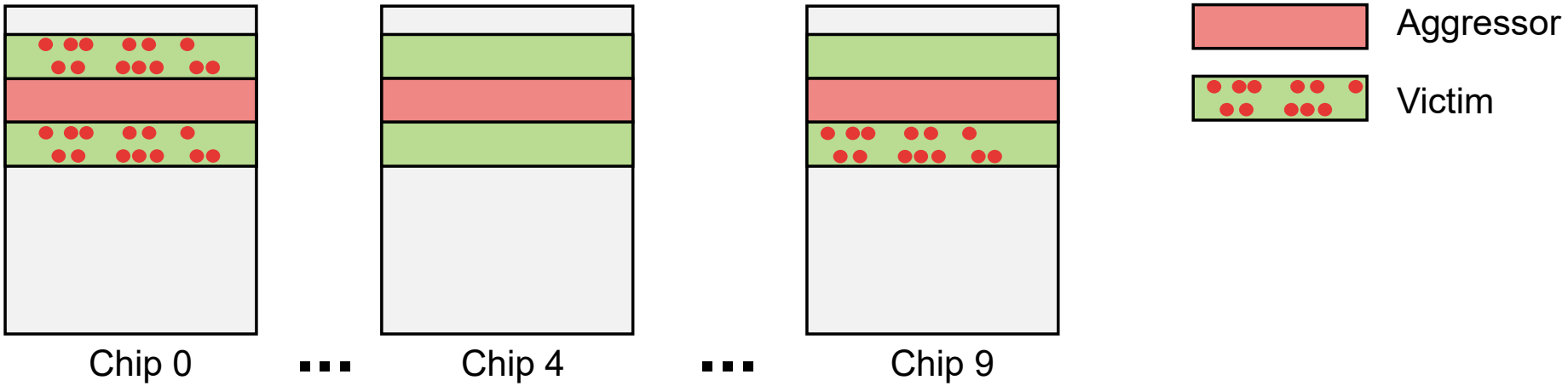
- 2) ECC corrects errors on memory access. Corrected values are **not written back** to DRAM.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-row scrambling

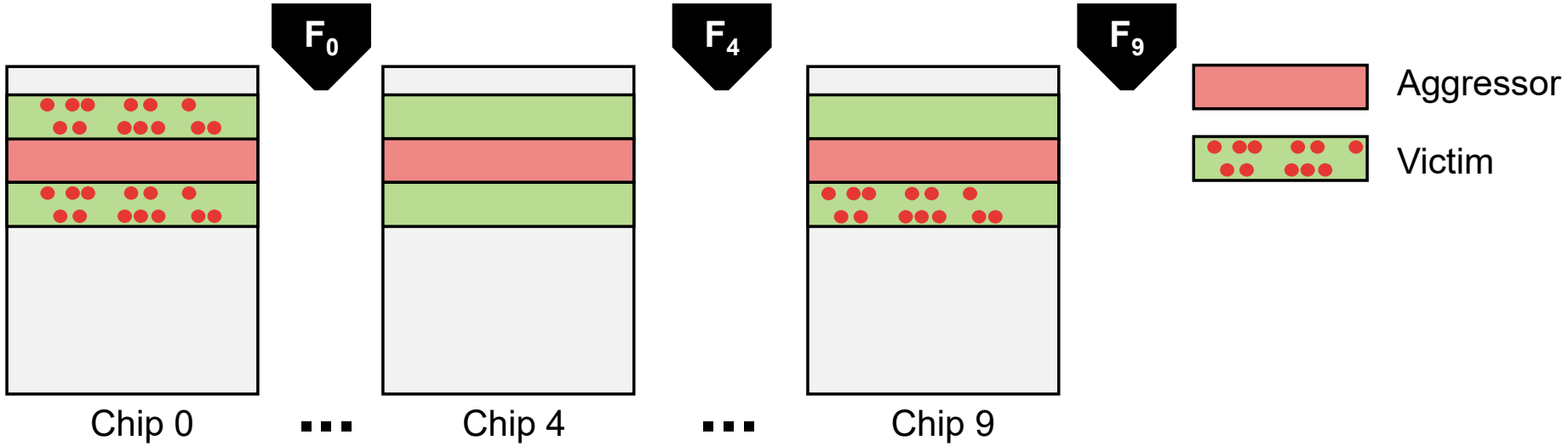
- Make every chip's victim unique using **per-chip row scrambling function (F)**.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-row scrambling

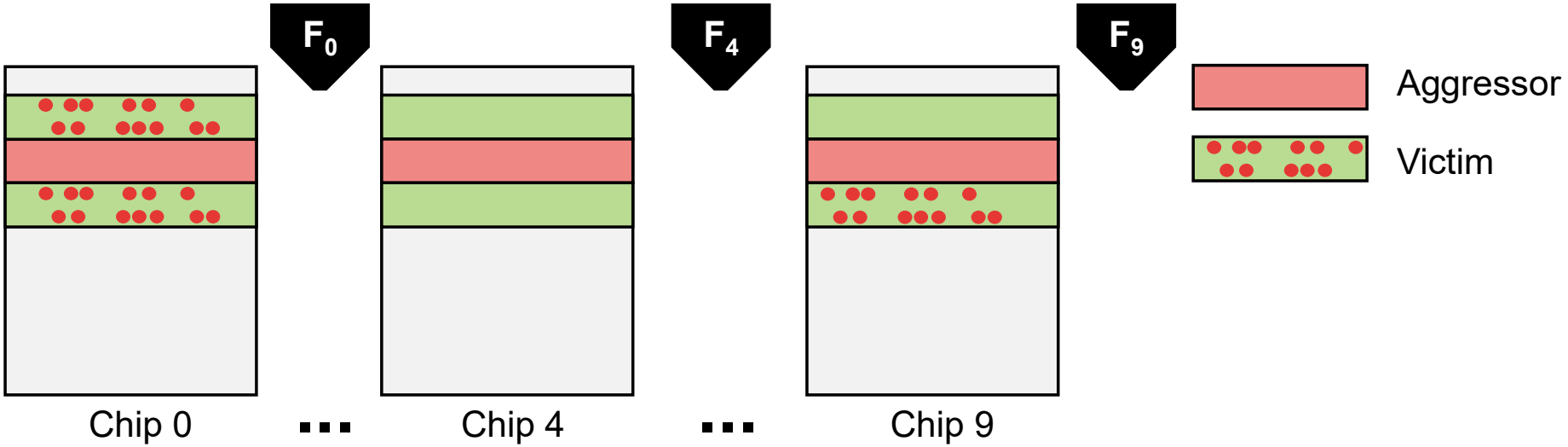
- Make every chip's victim unique using **per-chip row scrambling function (F)**.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-row scrambling

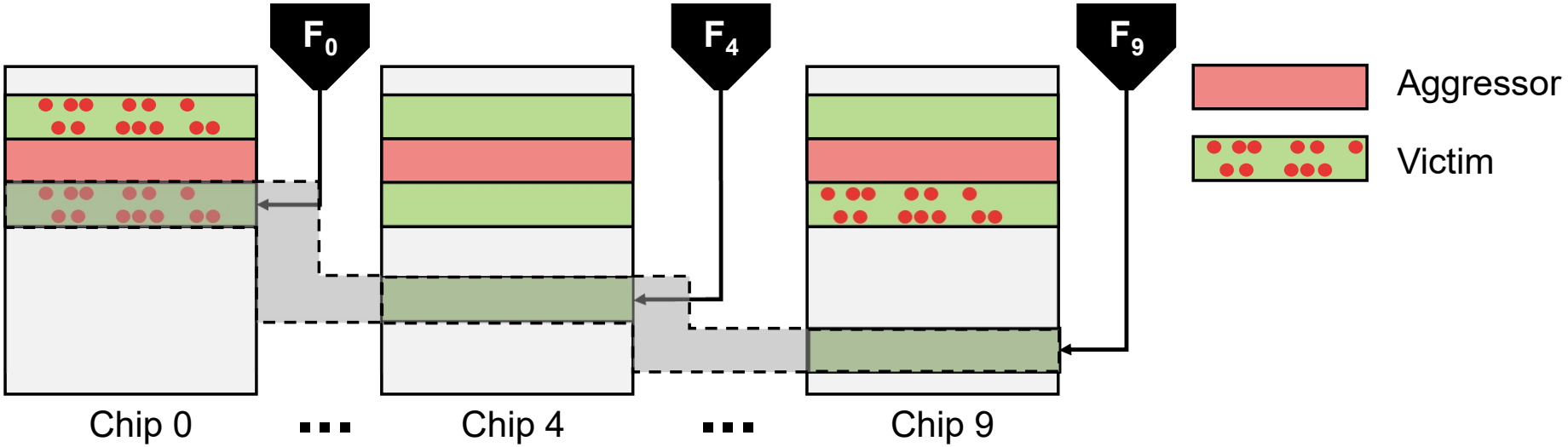
- Make every chip's victim unique using **per-chip row scrambling function (F)**.
  - Spread the victims to multiple Chipkill codewords.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-row scrambling

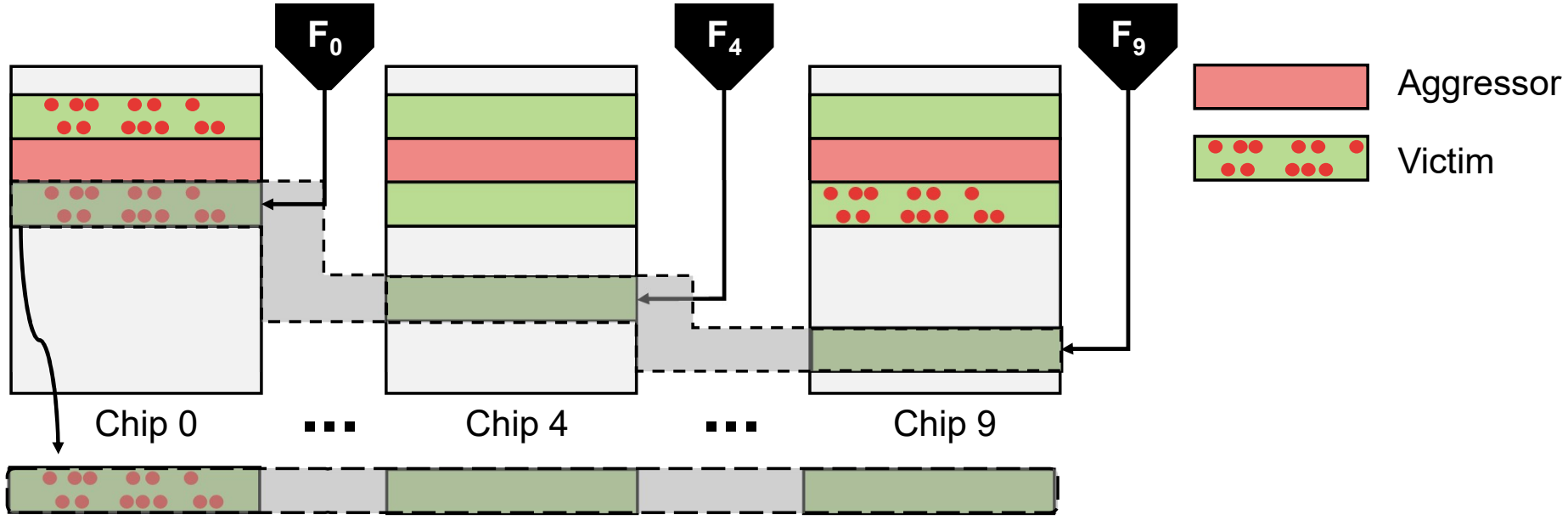
- Make every chip's victim unique using **per-chip row scrambling function (F)**.
  - Spread the victims to multiple Chipkill codewords.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-row scrambling

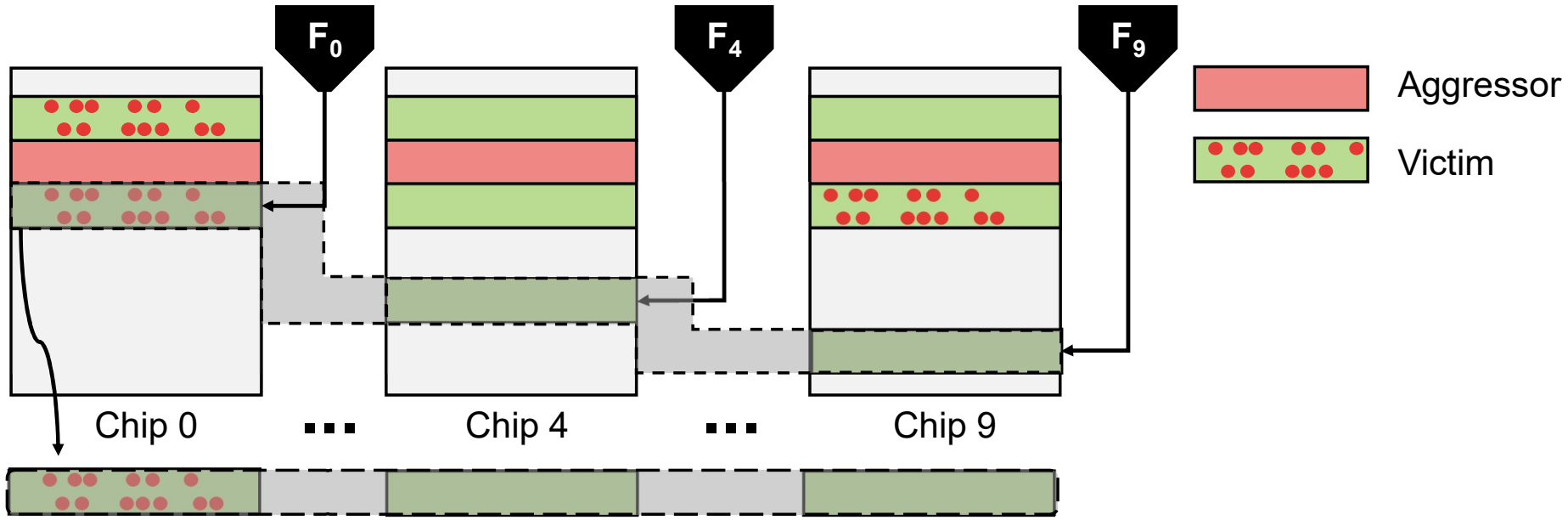
- Make every chip's victim unique using **per-chip row scrambling function (F)**.
  - Spread the victims to multiple Chipkill codewords.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]

- Chipkill correction timing side-channel enables **reverse-engineering** of the mapping function.

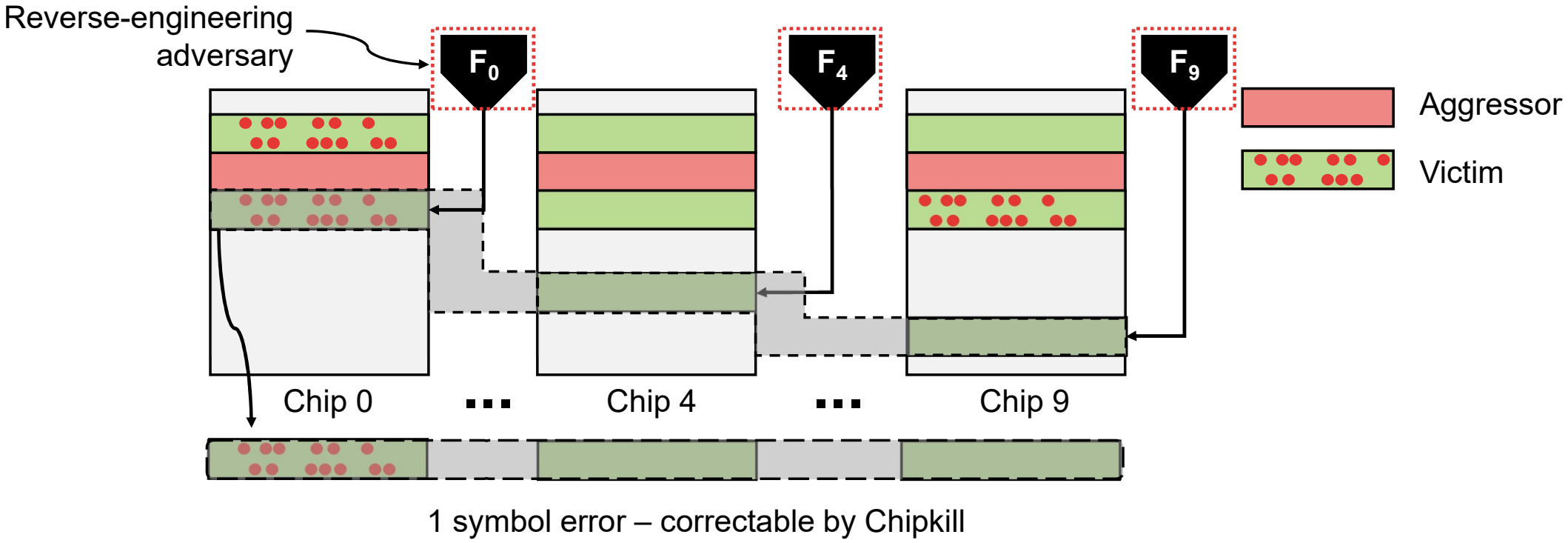


1 symbol error – correctable by Chipkill

[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]

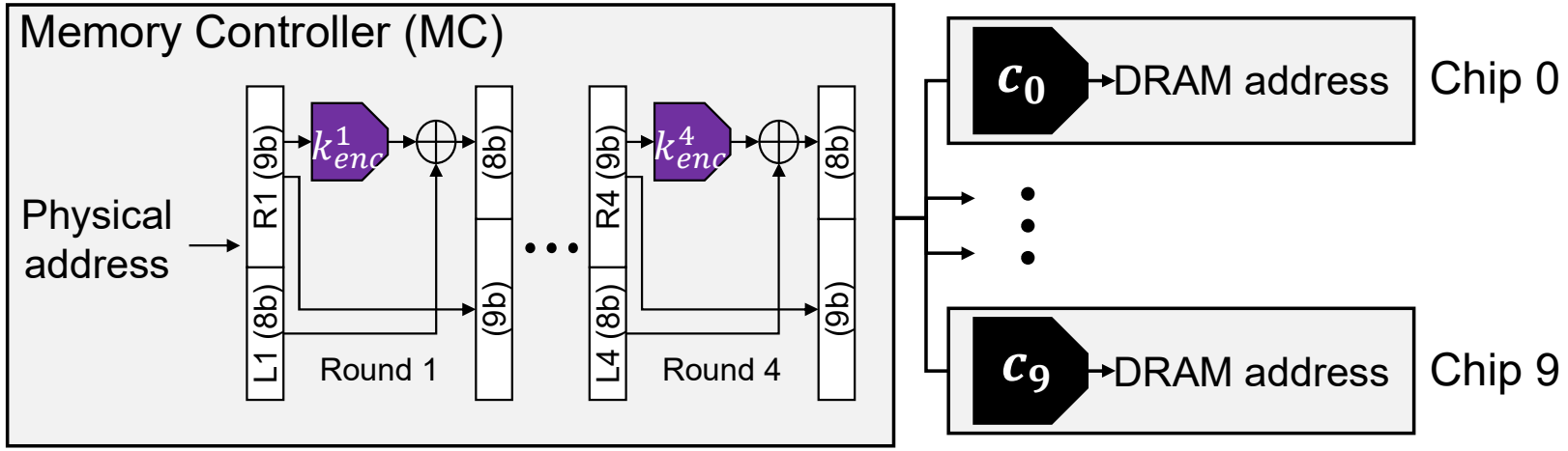
- Chipkill correction timing side-channel enables **reverse-engineering** of the mapping function.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-address obfuscation

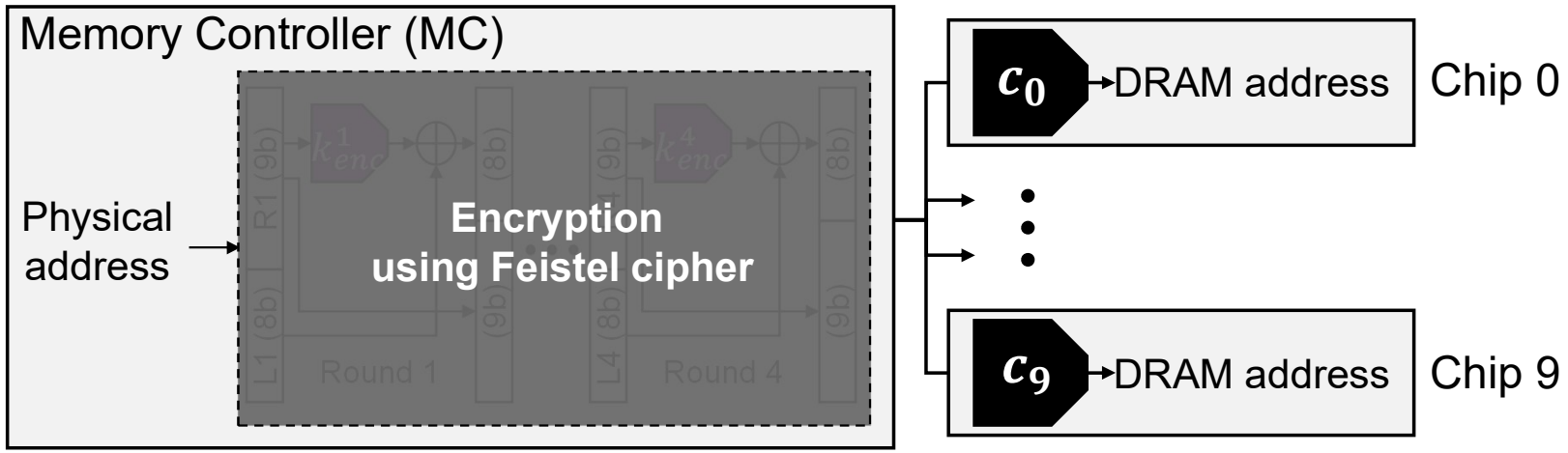
- Uses a **Feistel cipher** [2] protects the address scrambling function from reverse engineering.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.
[2] Black, John, and Phillip Rogaway. "Ciphers with arbitrary finite domains." *Cryptographers' track at the RSA conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

Cube [1]-address obfuscation

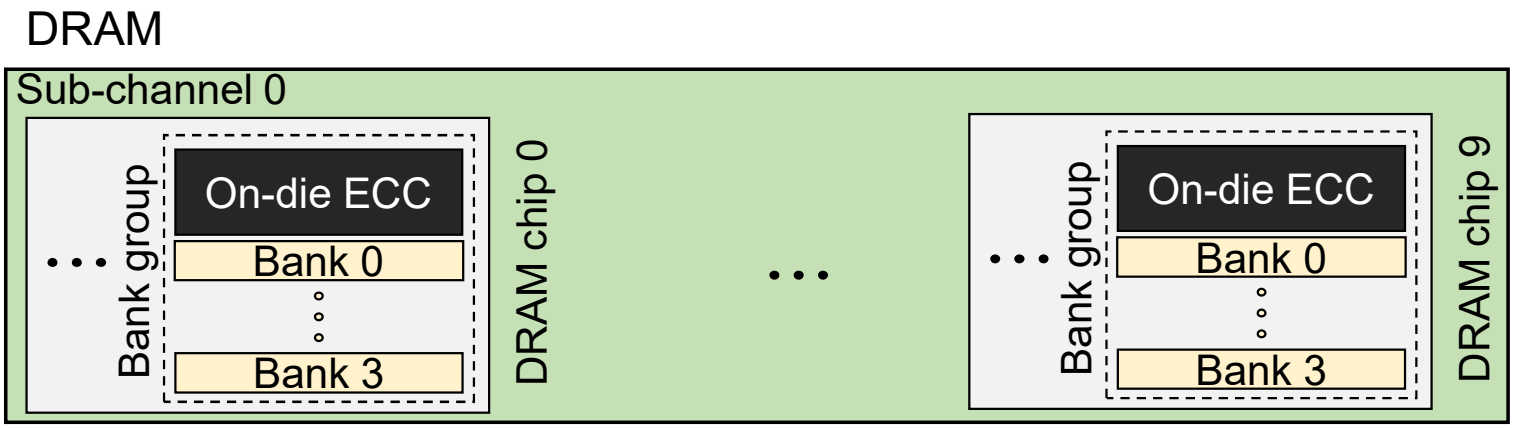
- Uses a **Feistel cipher** [2] protects the address scrambling function from reverse engineering.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.
[2] Black, John, and Phillip Rogaway. "Ciphers with arbitrary finite domains." *Cryptographers' track at the RSA conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

Cube [1]-scrubbing

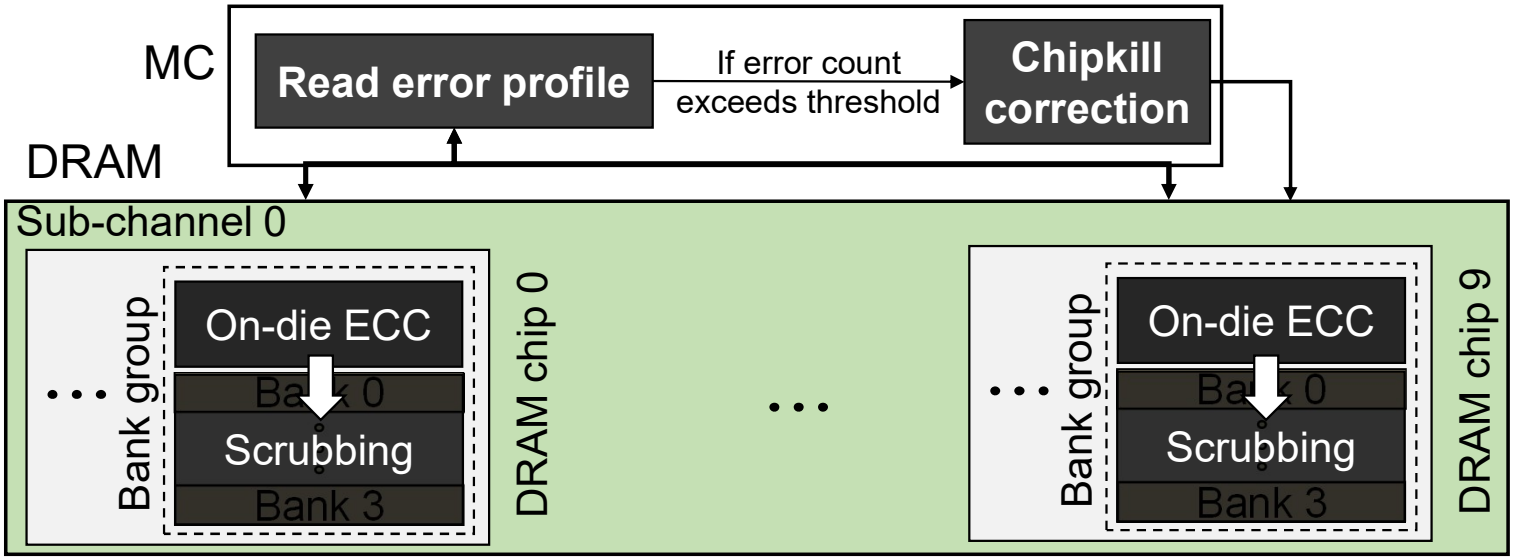
- Cube leverages OECC **scrubbing** (reads, corrects, and writes back clean data).



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Cube [1]-scrubbing

- Cube leverages OECC **scrubbing** (reads, corrects, and writes back clean data).
  - Detect RH victim rows before errors accumulate into an uncorrectable failure.
  - **Periodic scrubbing** with scrubbing window.



[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Limitations of Cube [1]

- Cube itself is primarily effective against **single-aggressor attacks**.
- Cube still **requires preventive mitigation** (e.g., victim row refresh, row swap) to prevent multi-aggressor attacks.

[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Limitations of Cube [1]

- Cube itself is primarily effective against **single-aggressor attacks**.
- Cube still **requires preventive mitigation** (e.g., victim row refresh, row swap) to prevent multi-aggressor attacks.

Existing reactive RH mitigation cannot reliably handle multi-aggressor attacks without preventive mitigations.

[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

RowArmor: A Robust Reactive Defense

RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.

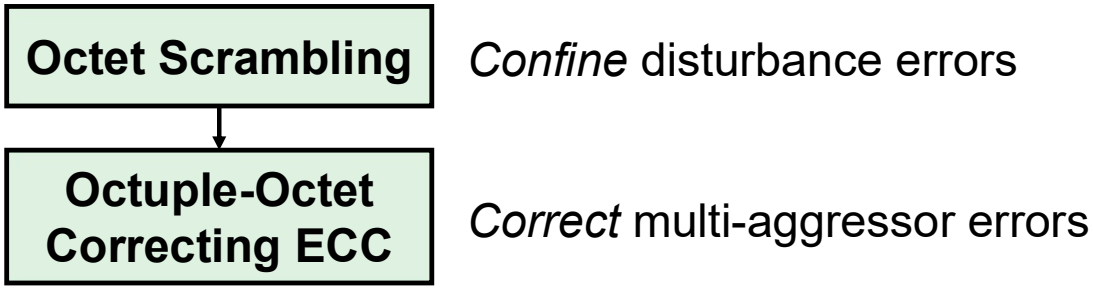
RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.

Octet Scrambling *Confine* disturbance errors

RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.

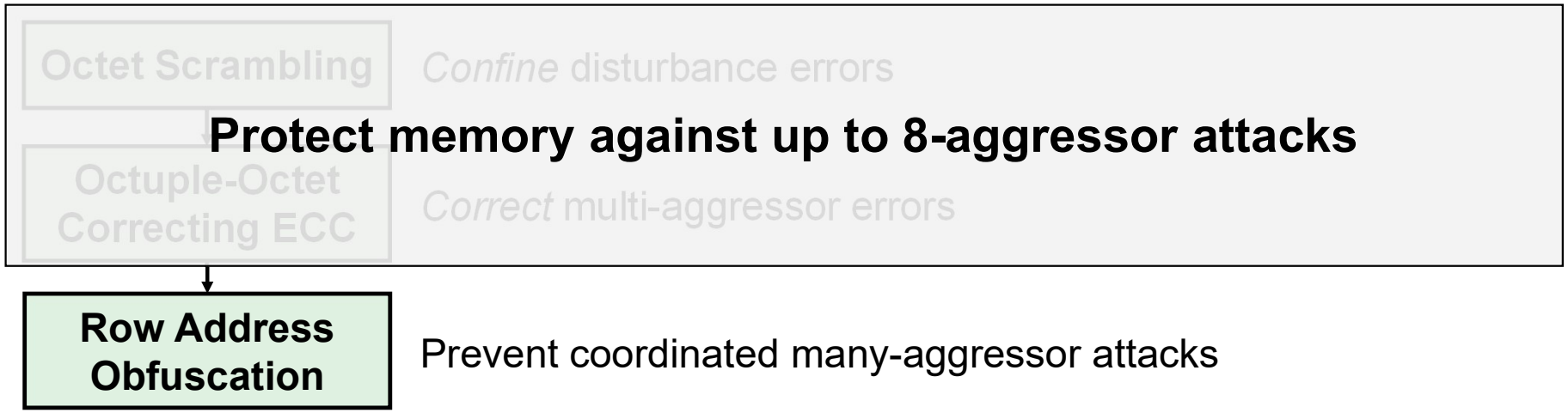


RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.

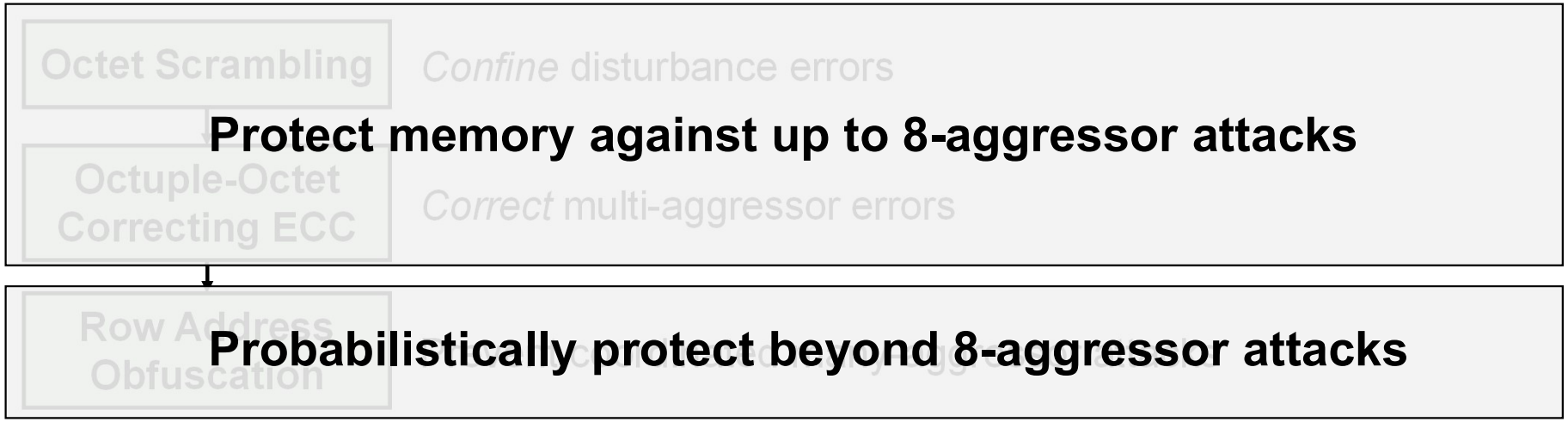
RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.



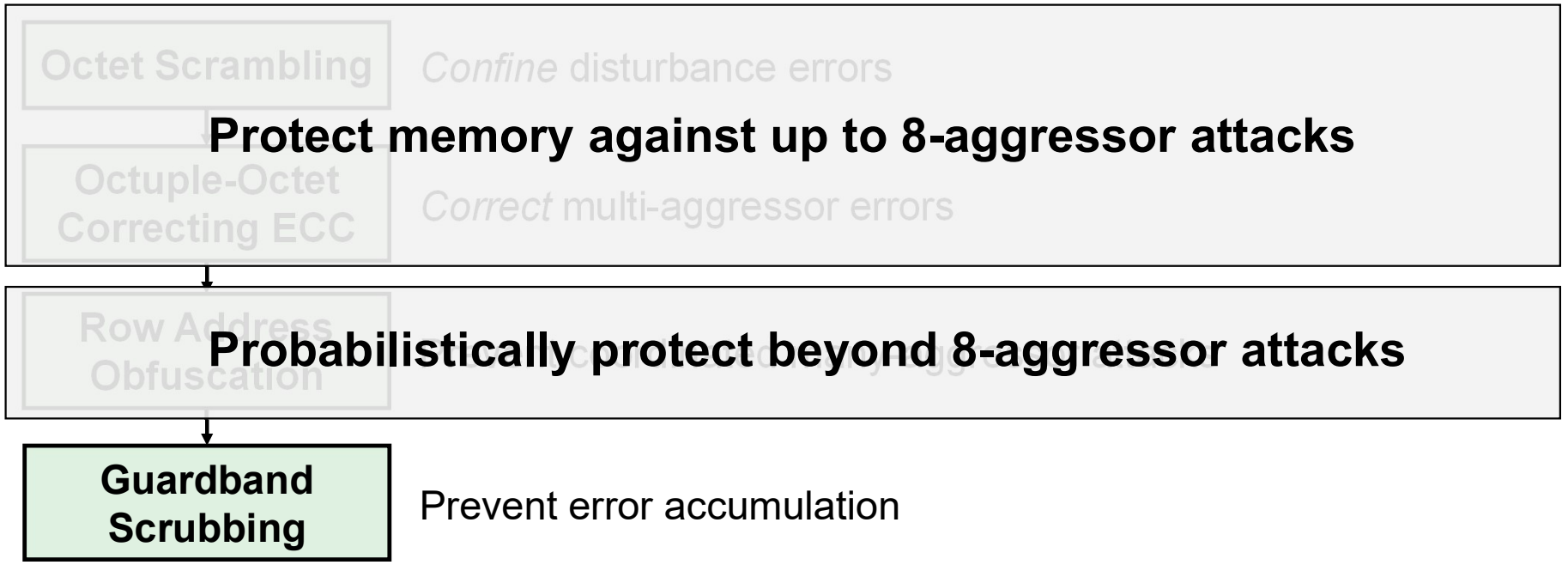
RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.



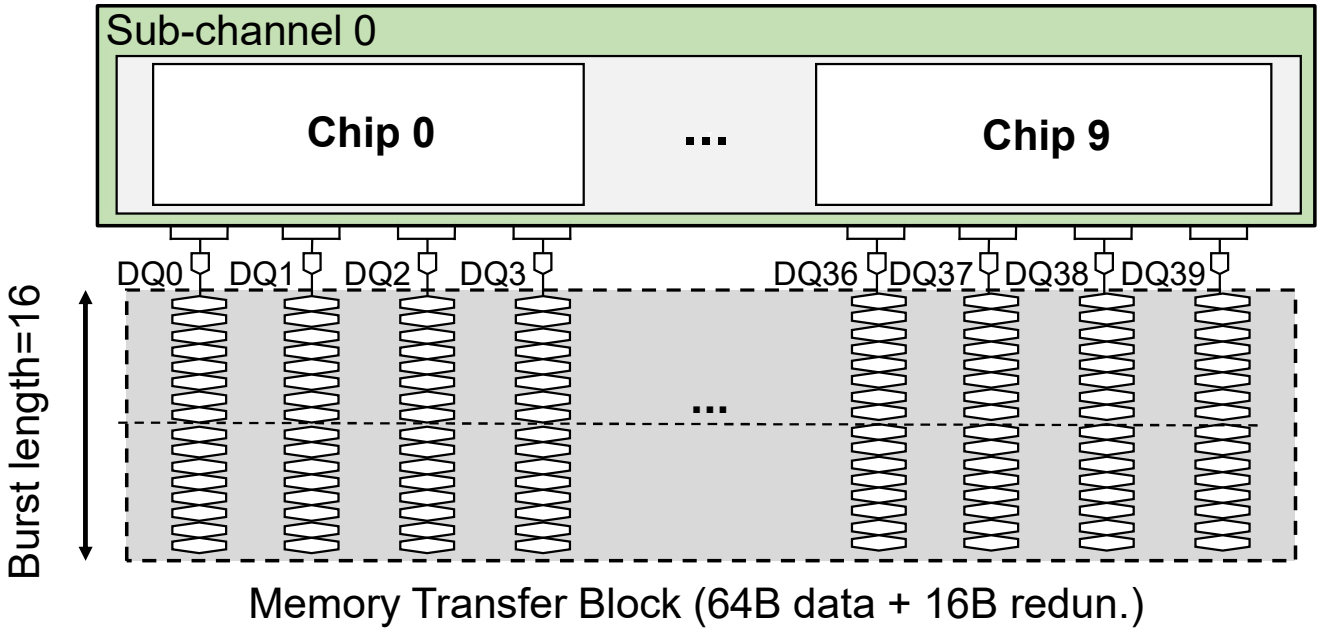
RowArmor: Overview

- RowArmor is a multi-layered reactive defense **built on Bamboo ECC and Cube**.
  - Enhanced ECC correction removes the need for preventive mitigation.



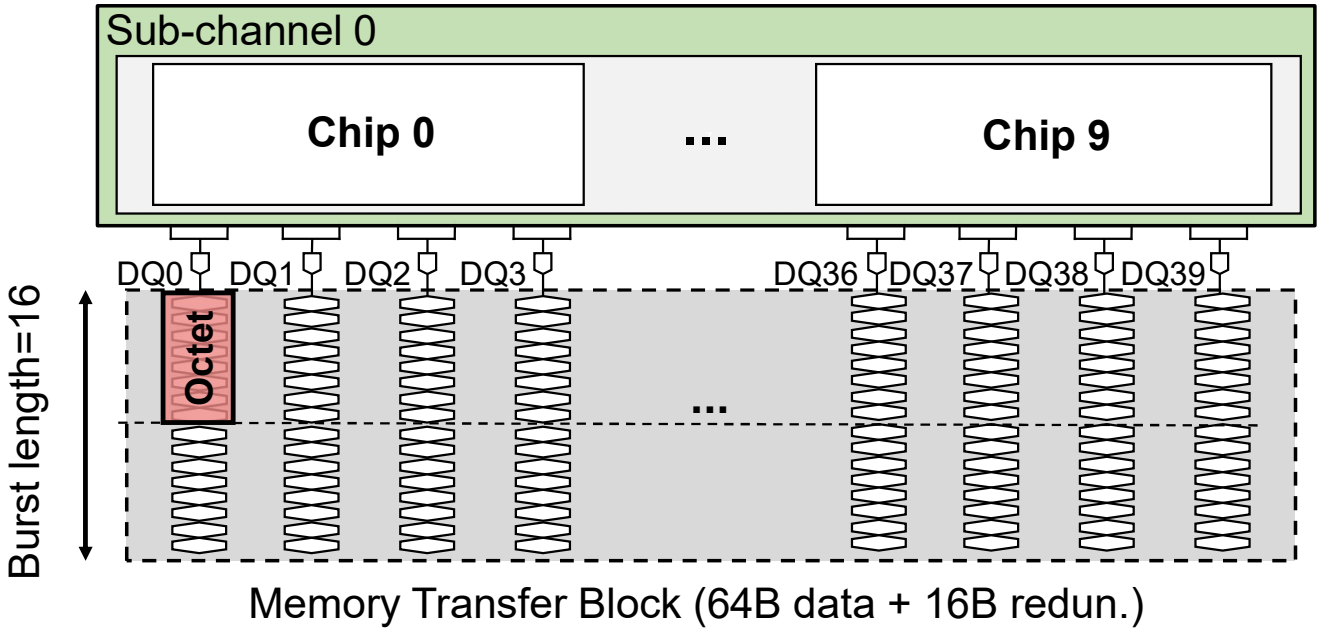
Octet scrambling

- Confines the impact of a single-aggressor RH attack to an 8-bit group (**octet**).
- Octet consists of bits from a **half-burst** transfer over a DQ.



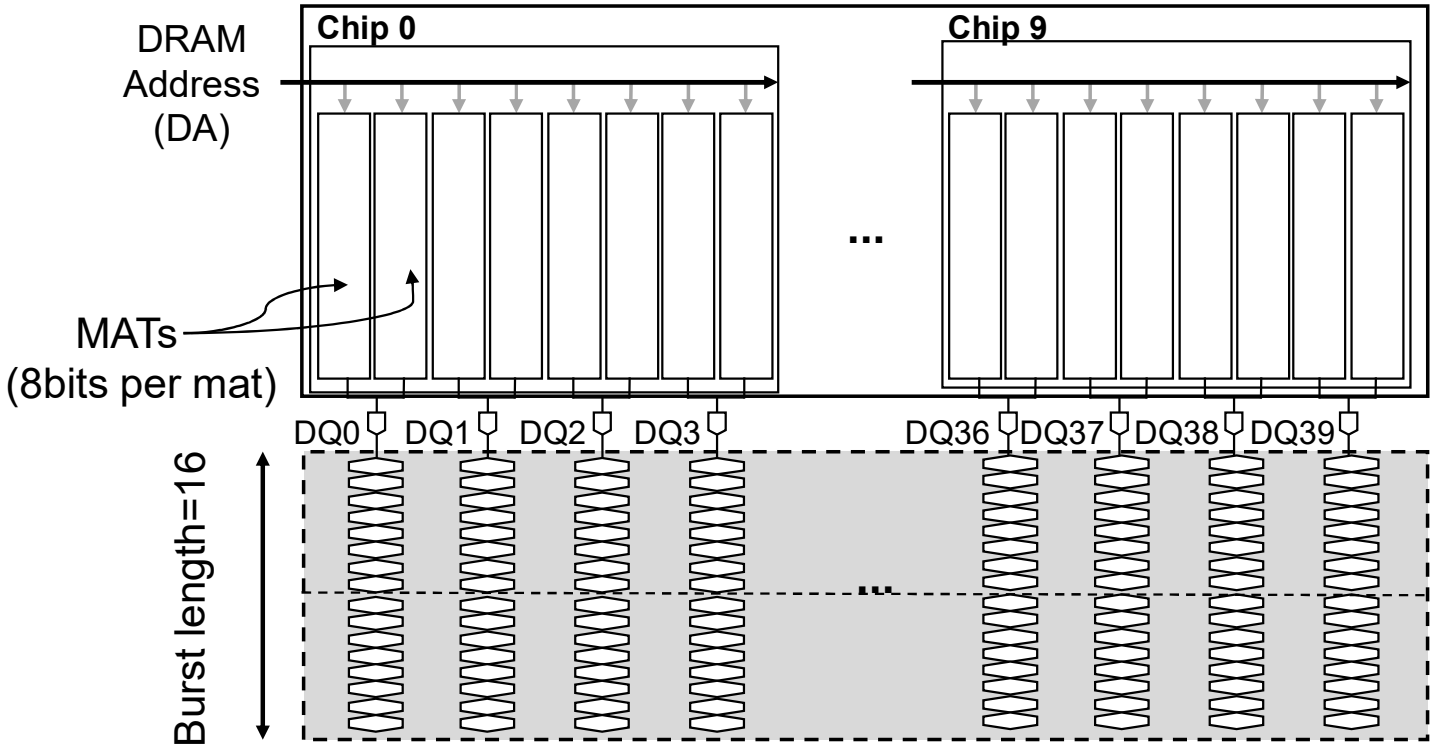
Octet scrambling

- Confines the impact of a single-aggressor RH attack to an 8-bit group (**octet**).
- Octet consists of bits from a **half-burst** transfer over a DQ.



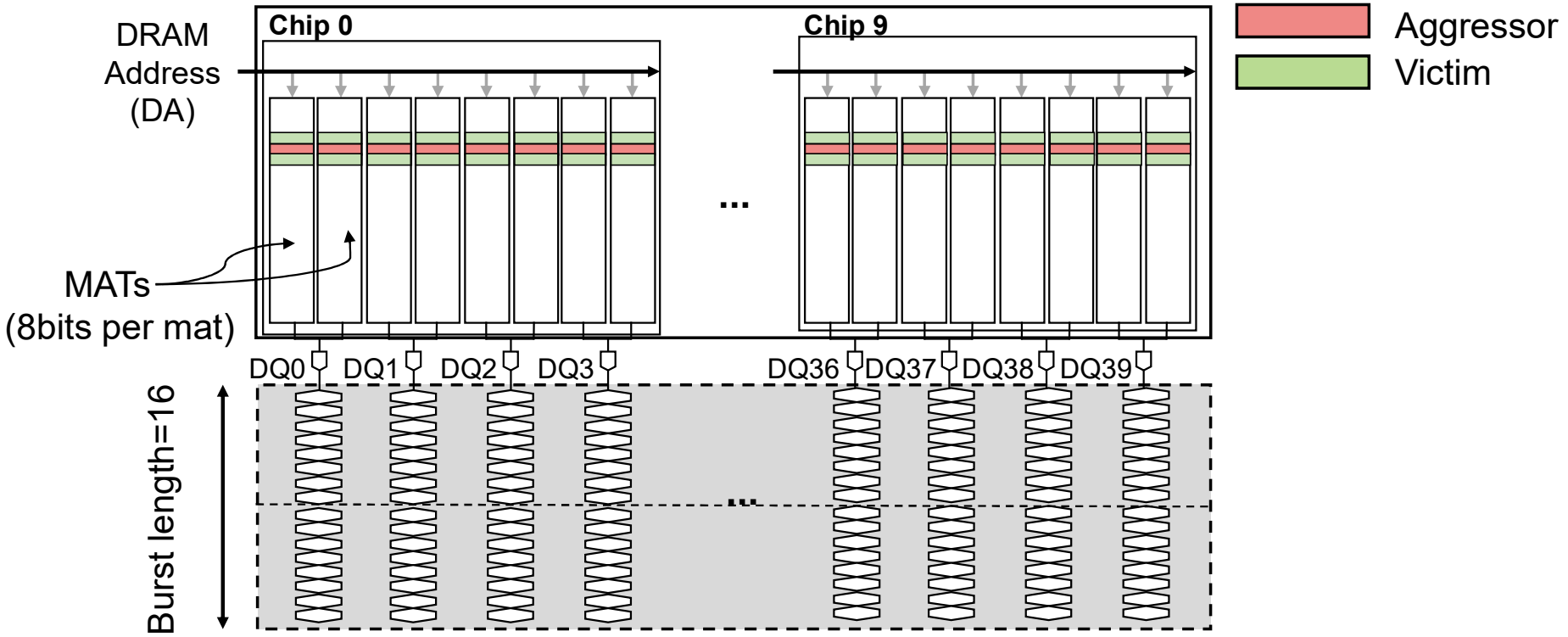
Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.



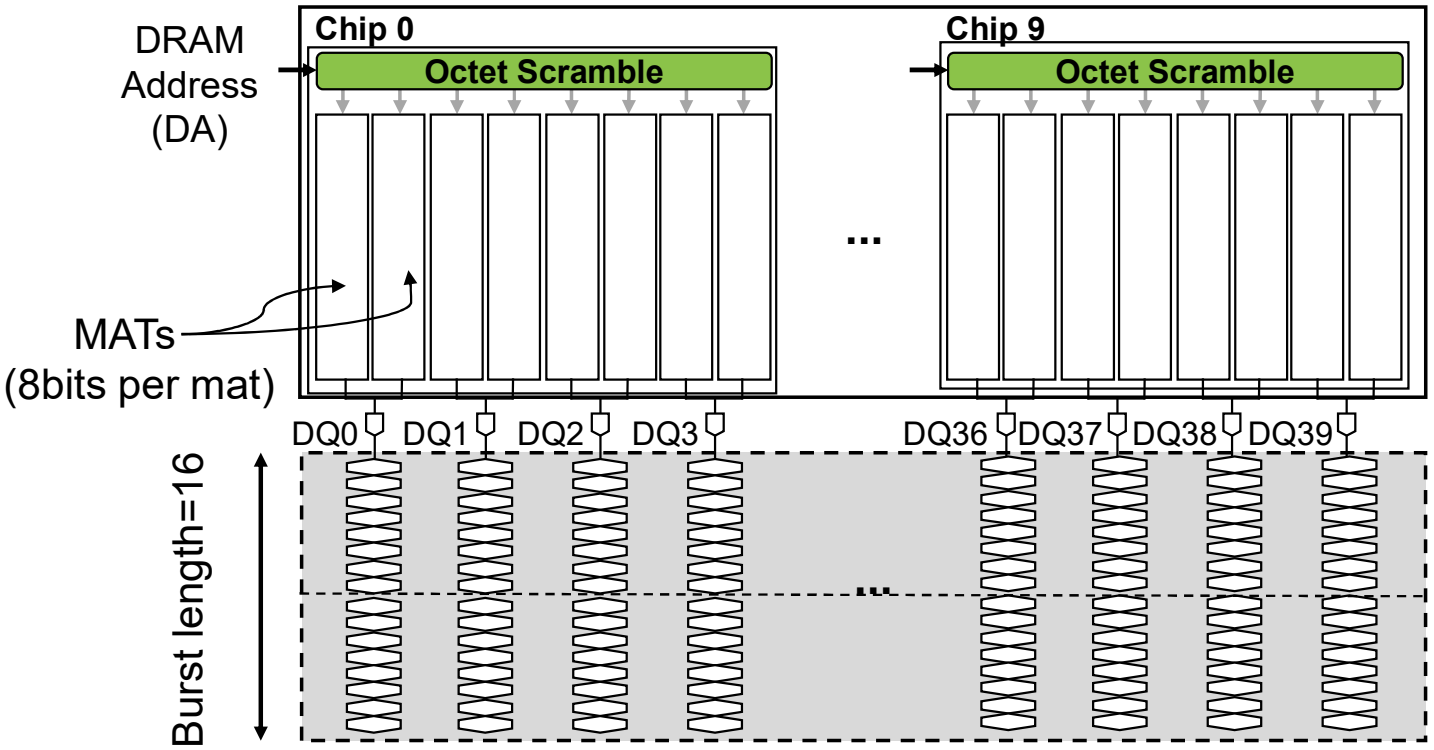
Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.



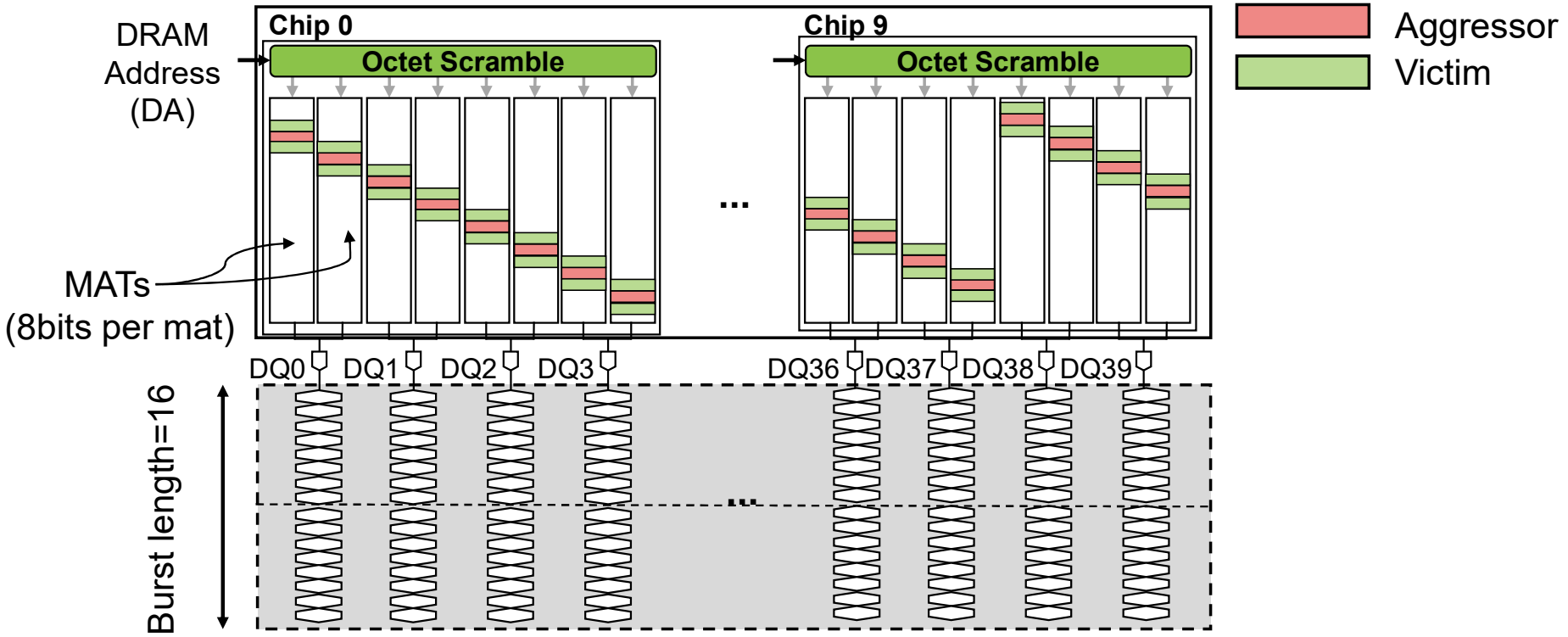
Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.



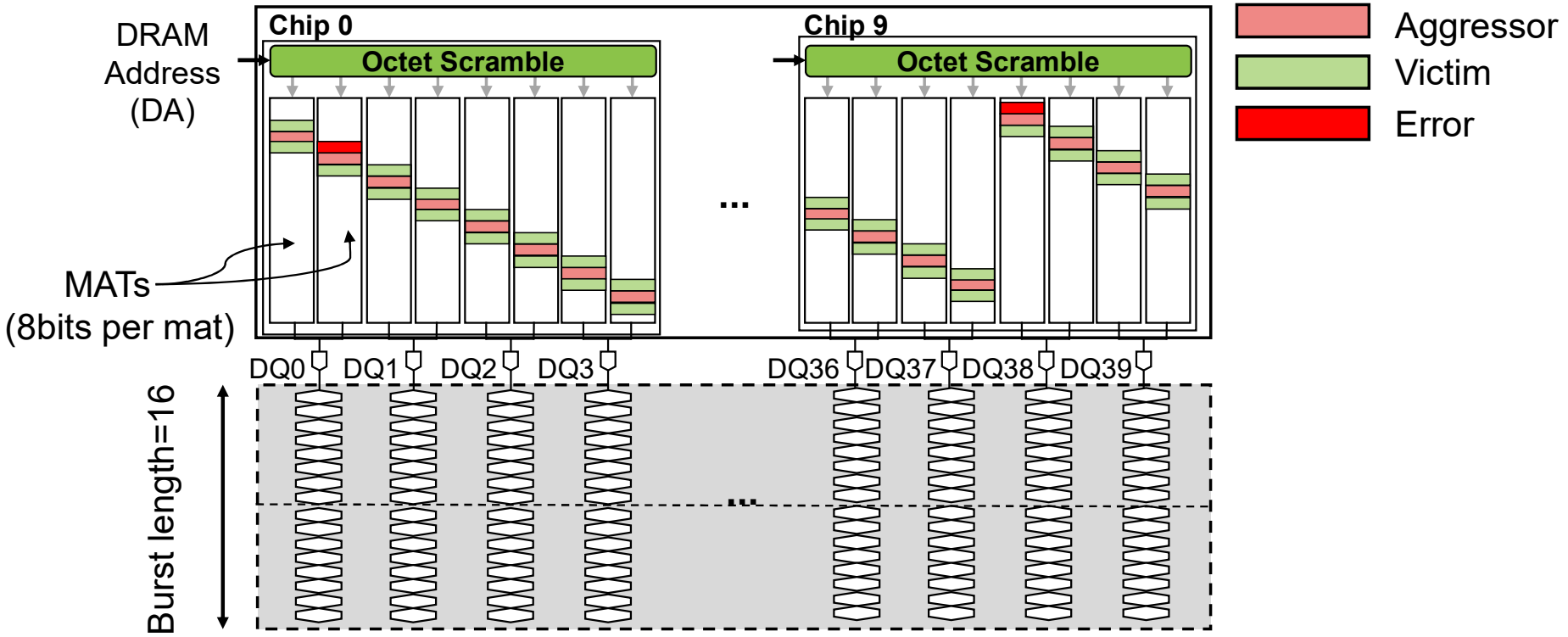
Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.



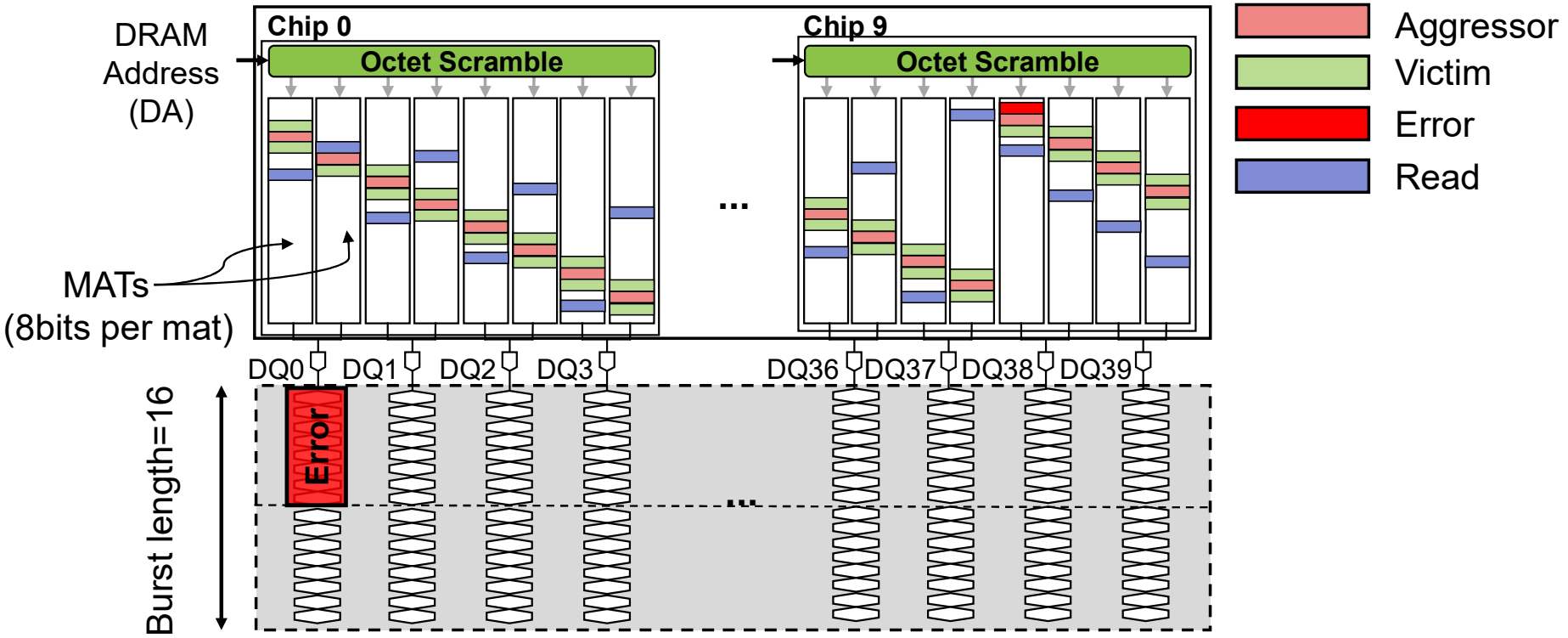
Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.



Octet scrambling

- Octet scrambling confines a single-aggressor attack to at most **one octet** per access.

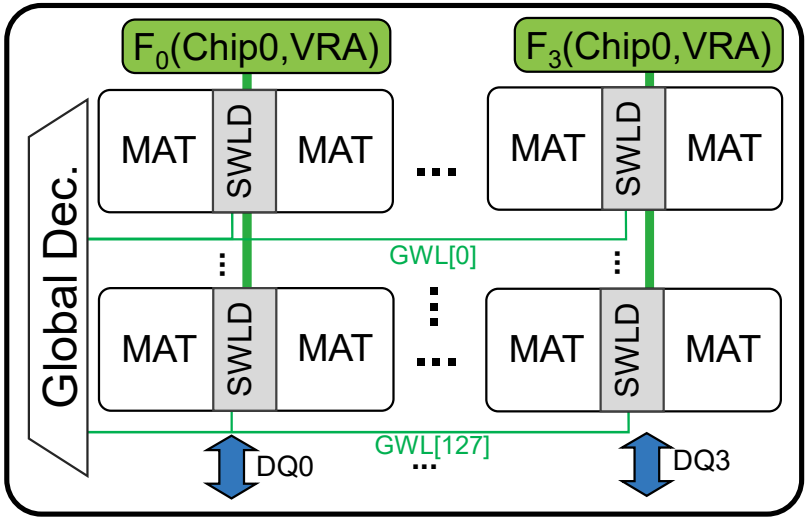


Octet scrambling

- Two mechanisms enforce octet-level confinement.
  - DQ Address Scrambling (**DAS**): Different address mappings across DQs and chips.
  - Sub-WordLine Permutation (**SWLP**): Different mappings between octets within a DQ.

Octet scrambling: DQ Address Scrambling (DAS)

- Extends the chip-level row address scrambling in Cube.
  - Differentiate address mappings across **DQs** and **DRAM chips.**

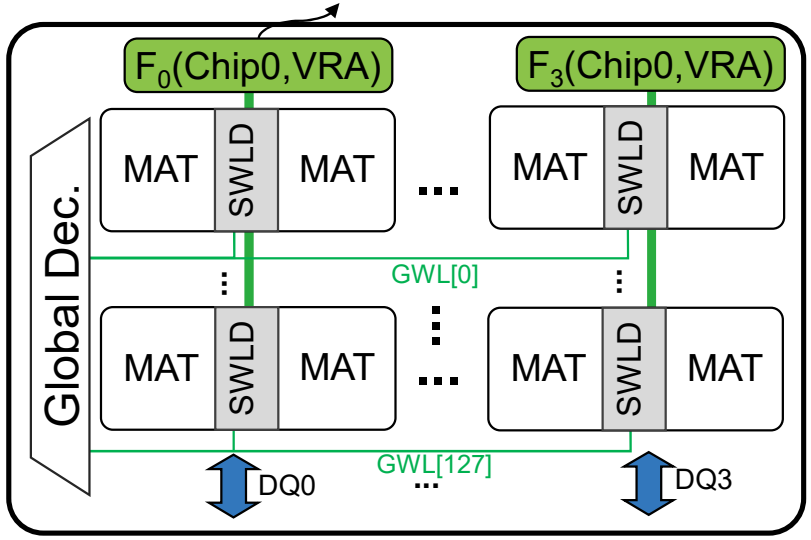


**Sub-WordLine Driver (SWLD)*
**Global Word Line (GWL)*

Octet scrambling: DQ Address Scrambling (DAS)

- Extends the chip-level row address scrambling in Cube.
  - Differentiate address mappings across **DQs** and **DRAM chips.**

Different scrambling Block for each DQs.

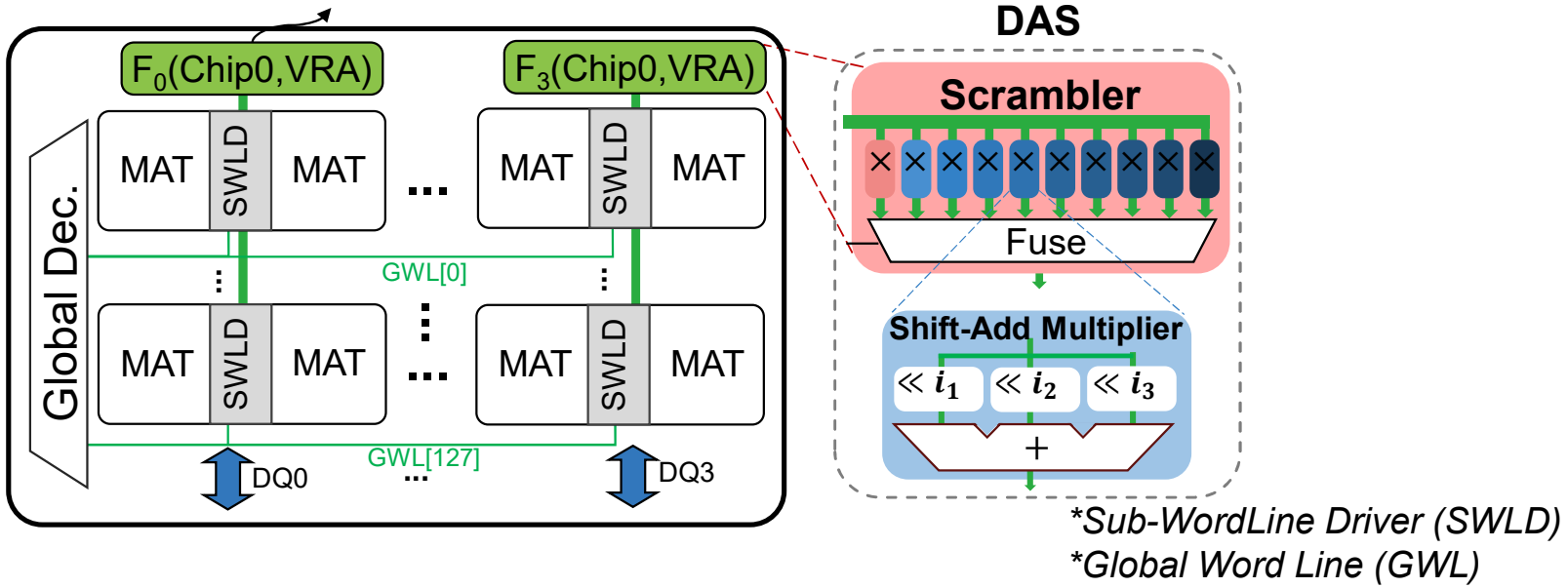


*Sub-WordLine Driver (SWLD)
*Global Word Line (GWL)

Octet scrambling: DQ Address Scrambling (DAS)

- Apply modular multiplication to row addresses.
  - $MA_i = (k_i \times DA)\ mod\ 1024$ (k_i : scrambling coefficient, MA : Mat Address, DA : Dram Address)
- Implemented efficiently using only adds and shifts.
  - Increases t_{RCD} by only 1 cycle.

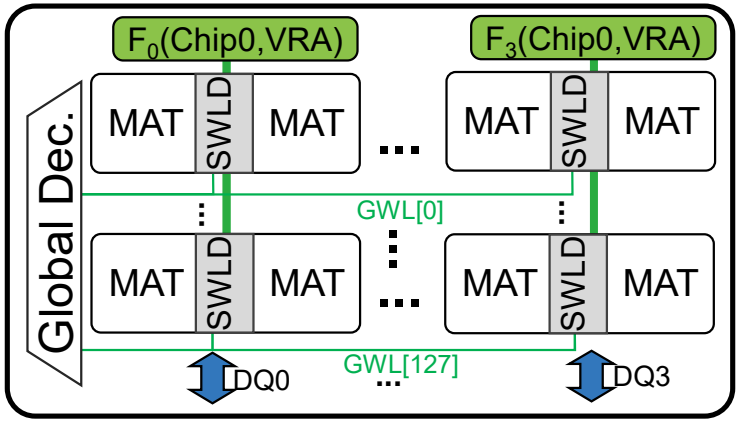
Different scrambling Block for each DQs.



*Sub-WordLine Driver (SWLD)
*Global Word Line (GWL)

Octet scrambling: Sub-WordLine Permutation (SWLP)

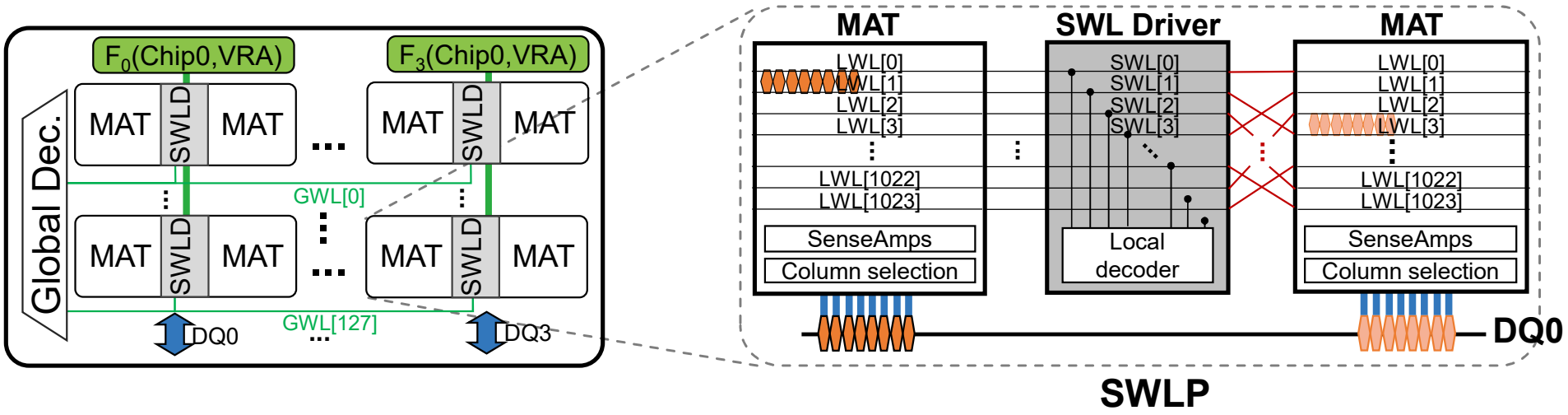
- Prevent two octets in the **same DQ** from being corrupted by one aggressor.



**Sub-WordLine Driver (SWLD)*
**Global Word Line (GWL)*

Octet scrambling: Sub-WordLine Permutation (SWLP)

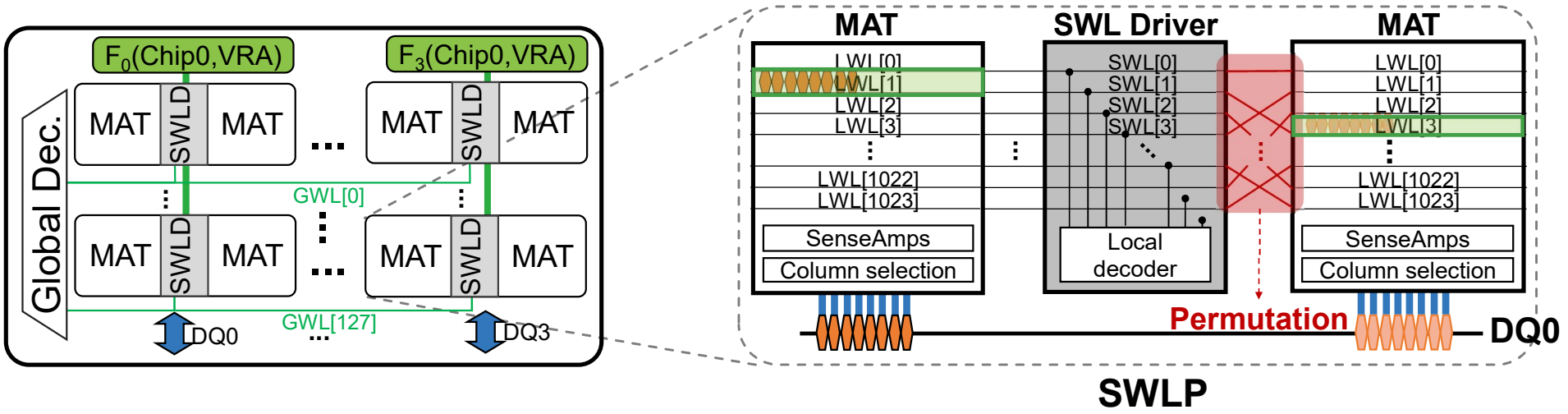
- Prevent two octets in the **same DQ** from being corrupted by one aggressor.



**Sub-WordLine Driver (SWLD)*
**Global Word Line (GWL)*
**Local Word Line (LWL)*

Octet scrambling: Sub-WordLine Permutation (SWLP)

- Prevent two octets in the **same DQ** from being corrupted by one aggressor.
- Requires only wire permutation (no hardware overhead).



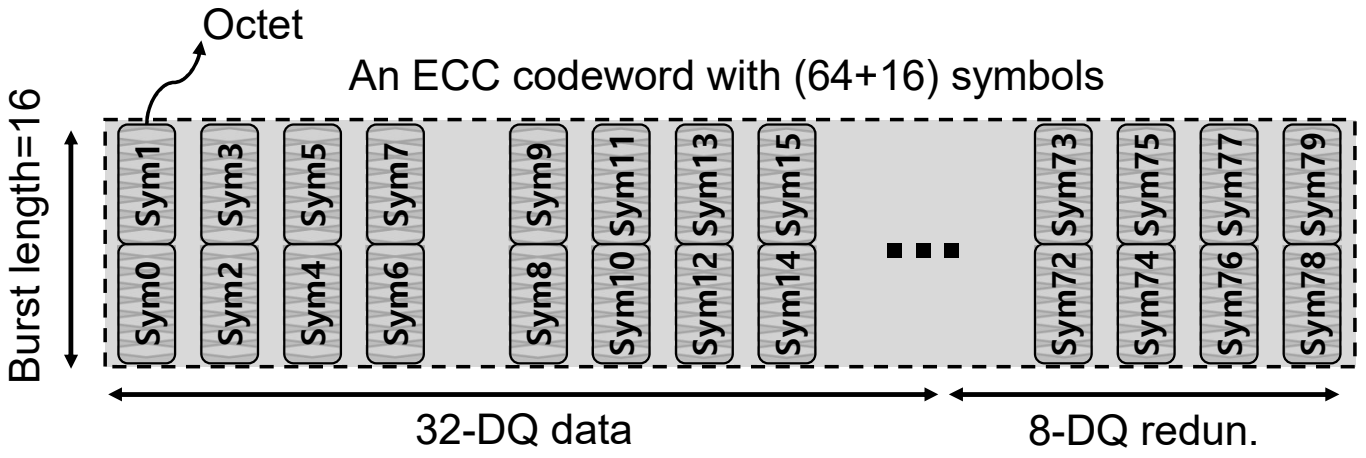
**Sub-WordLine Driver (SWLD)*
**Global Word Line (GWL)*
**Local Word Line (LWL)*

Octuple-Octet Correcting (OOC) ECC

- Octet Scrambling protects against single-aggressor attacks.
  - Correctable by Chipkill.
- Multi-aggressor attacks can create overlapping errors, exceeding conventional ECC capability.
  - Needs **stronger ECC** that matches new data layout (octet).

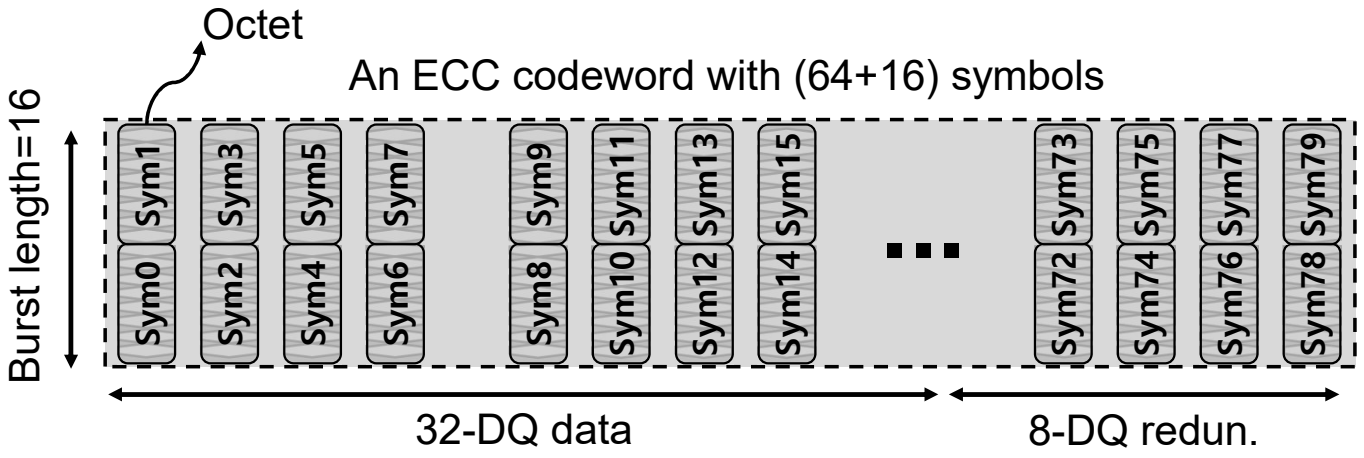
Octuple-Octet Correcting (OOC) ECC

- Design inspired by QPC of Bamboo ECC.
  - 64 data symbols + 16 parity symbols (**8 symbol correct**)
- Match symbol to octet-level errors.



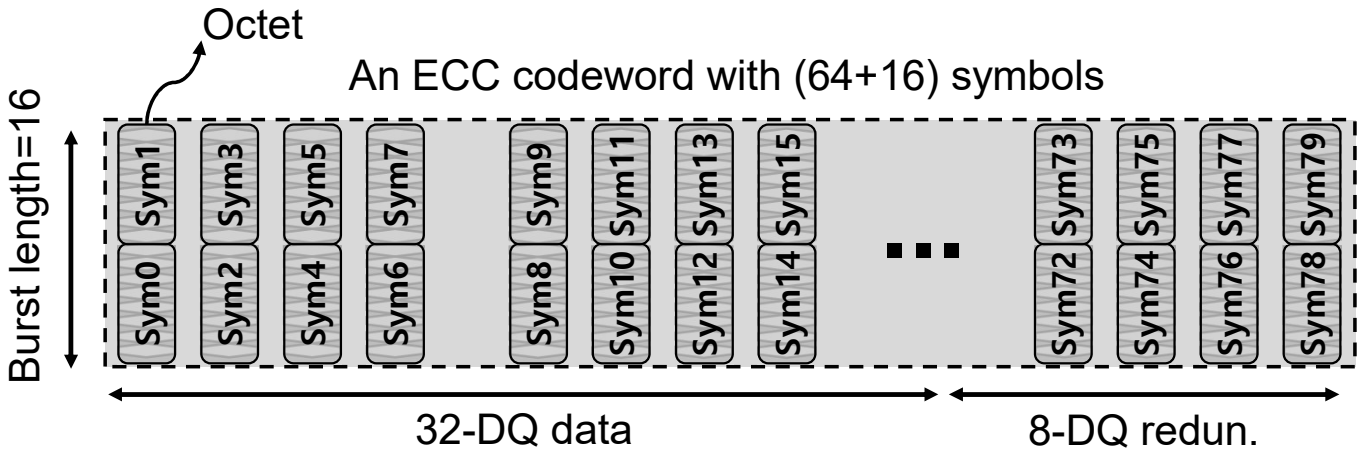
Octuple-Octet Correcting (OOC) ECC

- **Correct up to eight octet errors.**
  - Protect against 8 aggressor attacks.
- Requires the full block for decoding.
  - Increases read latency by only 1 cycle (due to DDR5 PHY 8:1 deserialization).



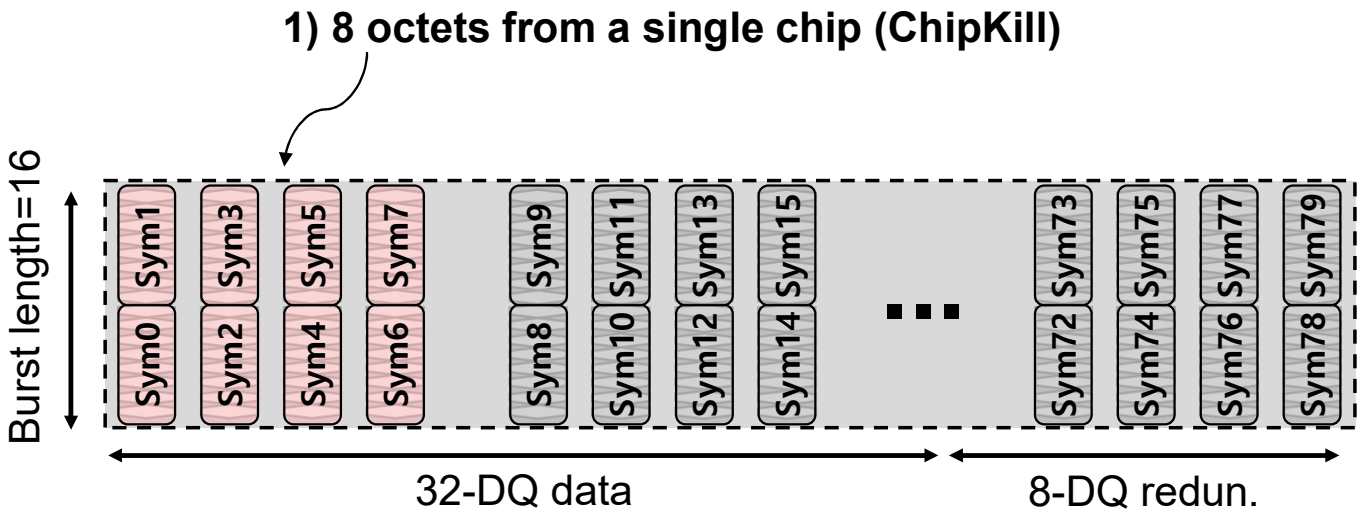
Octuple-Octet Correcting (OOC) ECC

- Validate whether errors are due to RH disturbance or random faults.
- **Correction Validation** balances strong error correction for safety with miscorrection prevention for reliability.



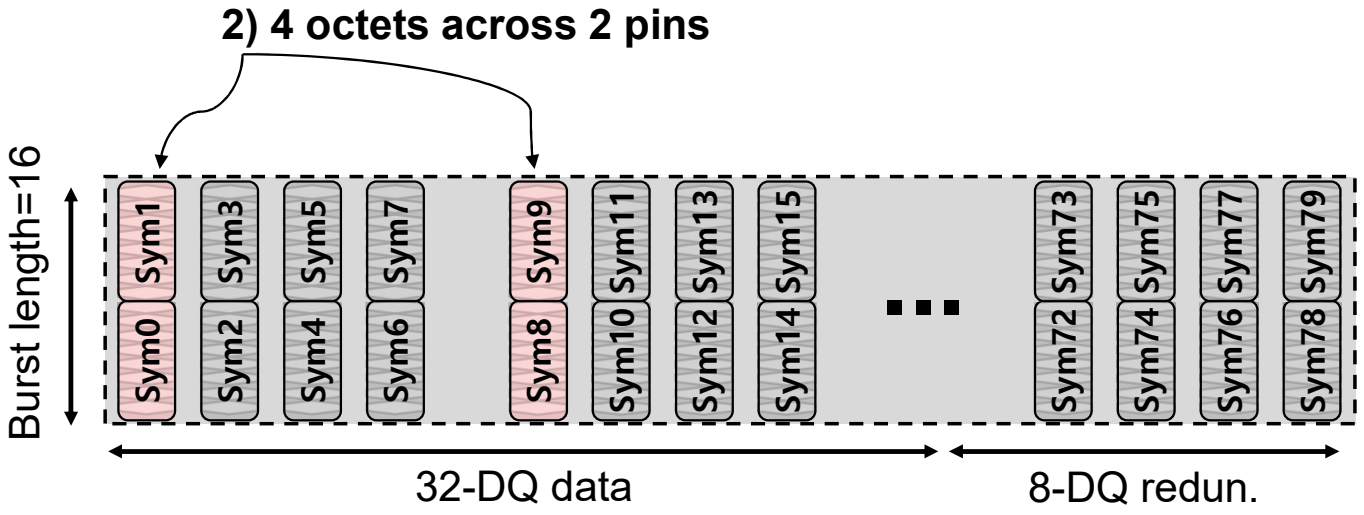
Octuple-Octet Correcting (OOC) ECC

- Trigger validation when the specific error patterns occur.



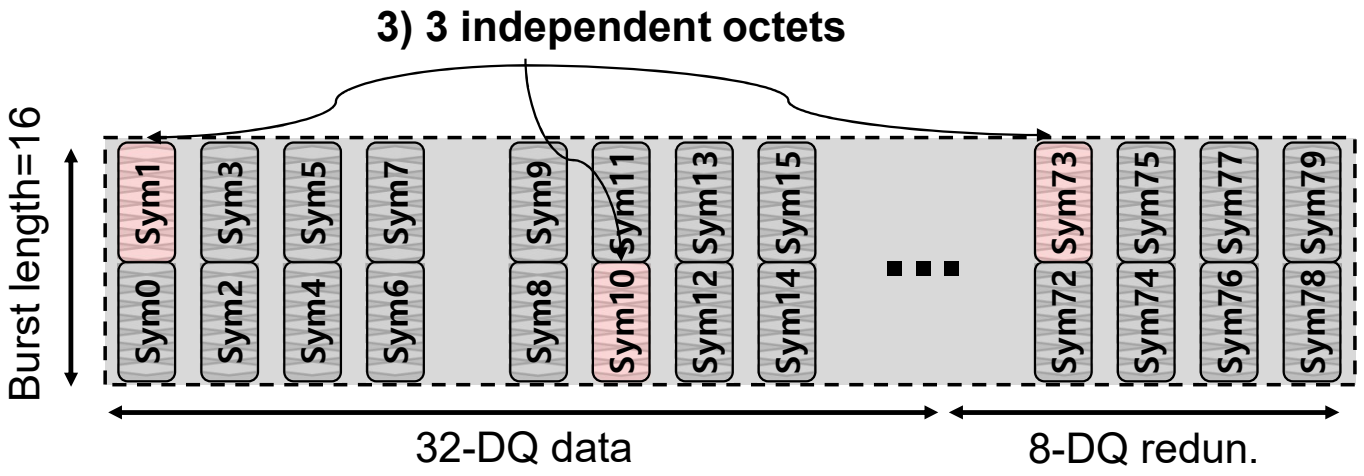
Octuple-Octet Correcting (OOC) ECC

- Trigger validation when the specific error patterns occurred.



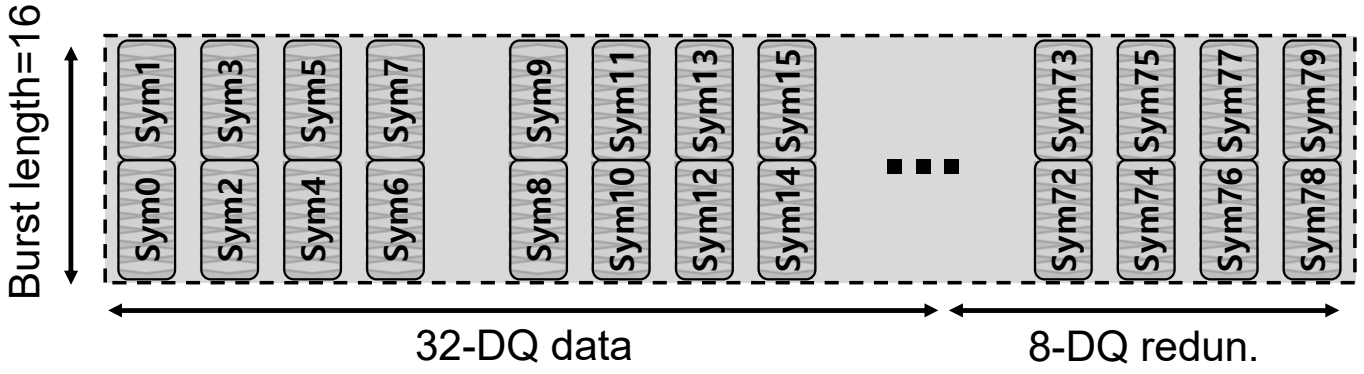
Octuple-Octet Correcting (OOC) ECC

- Trigger validation when the specific error patterns occurred.



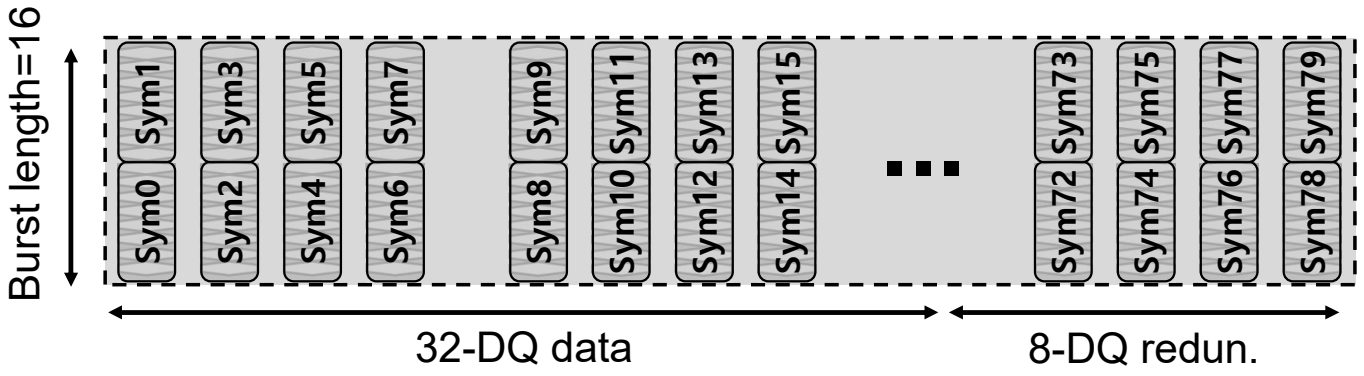
Octuple-Octet Correcting (OOC) ECC

- Trigger validation when the specific error patterns occurred.
  - Read OECC error counters and perform additional same-row read.



Octuple-Octet Correcting (OOC) ECC

- Trigger validation when the specific error patterns occurred.
  - Read OECC error counters and perform additional same-row read.
  - If counter increases, accept correction (RH disturbance error).



Beyond 8-aggressor attacks

- Octet scrambling and OOC ECC: Protects against **up to 8 aggressors**

Beyond 8-aggressor attacks

- Octet scrambling and OOC ECC: Protects against **up to 8 aggressors**
- Attacker may still attempt:
  - Increase the number of aggressor rows (inducing more than 8 overlapping errors).
  - Exceed OOC ECC correction capability.

Row Address Obfuscation (RAO)

- Randomizes the mapping between physical and DRAM row addresses
- Per-bank secret keys encrypt row addresses using a **Feistel-cipher** [1] permutation.
  - Implemented in the MC (same as Cube).

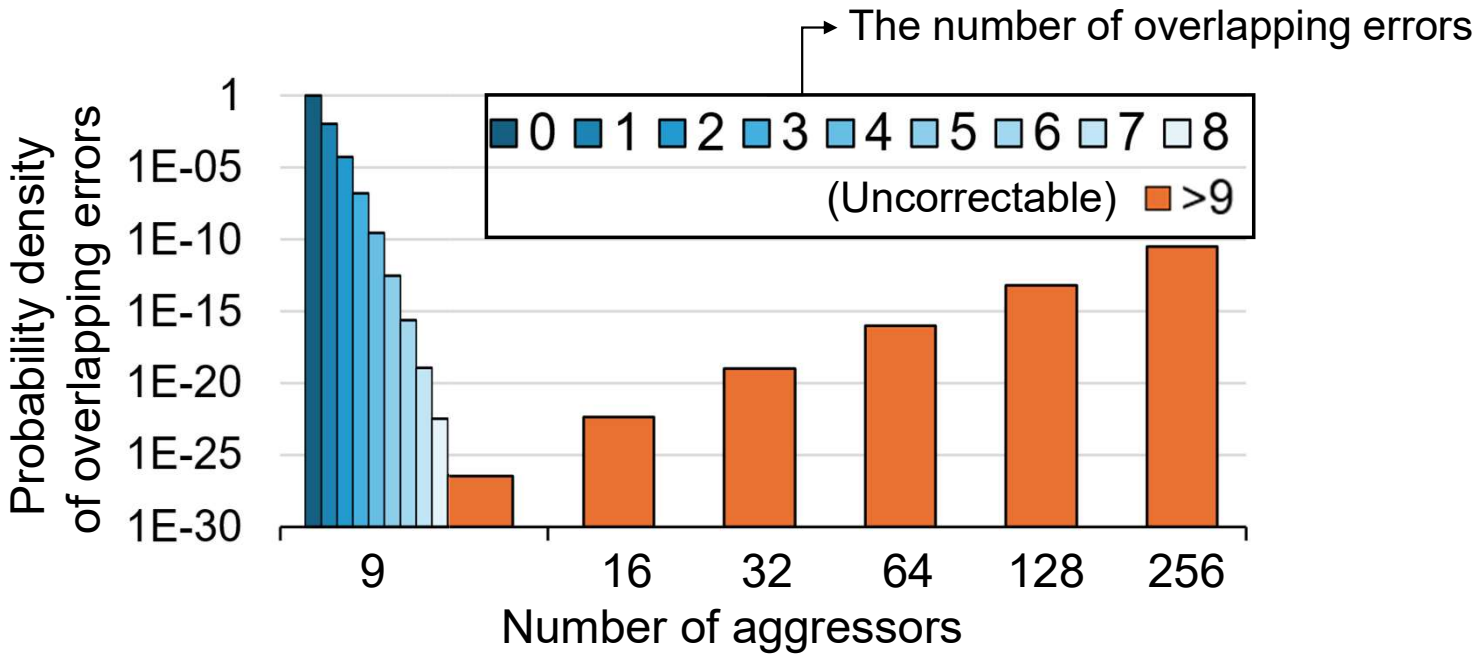
[1] Black, John, and Phillip Rogaway. "Ciphers with arbitrary finite domains." *Cryptographers' track at the RSA conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

Row Address Obfuscation (RAO)

- Effectiveness of **RAO**

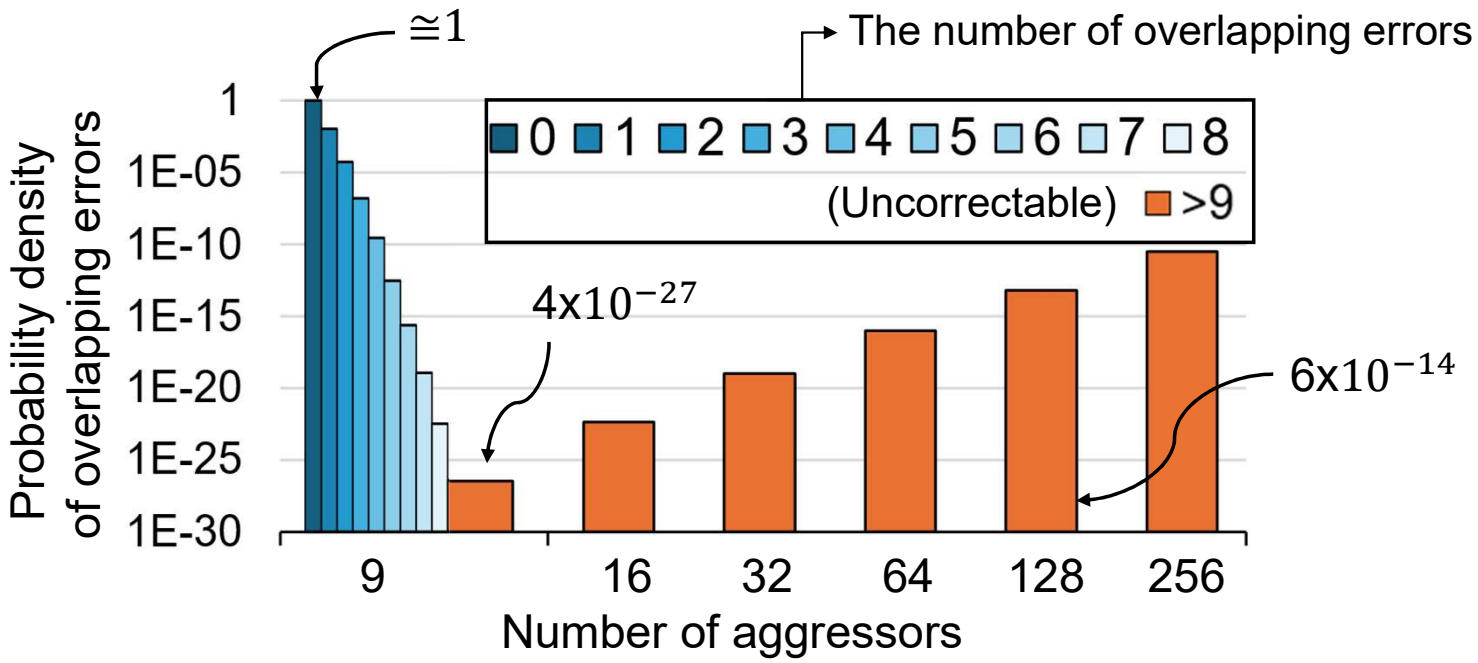
Row Address Obfuscation (RAO)

- Effectiveness of **RAO**



Row Address Obfuscation (RAO)

- Effectiveness of **RAO**



Preventing error accumulation

- OOC ECC corrects errors on memory access but corrected values are not written back.
  - Errors in unaccessed rows remain and **accumulate** over time.
- Accumulated errors may exceed the 8-octet correction capability.

Guardband scrubbing

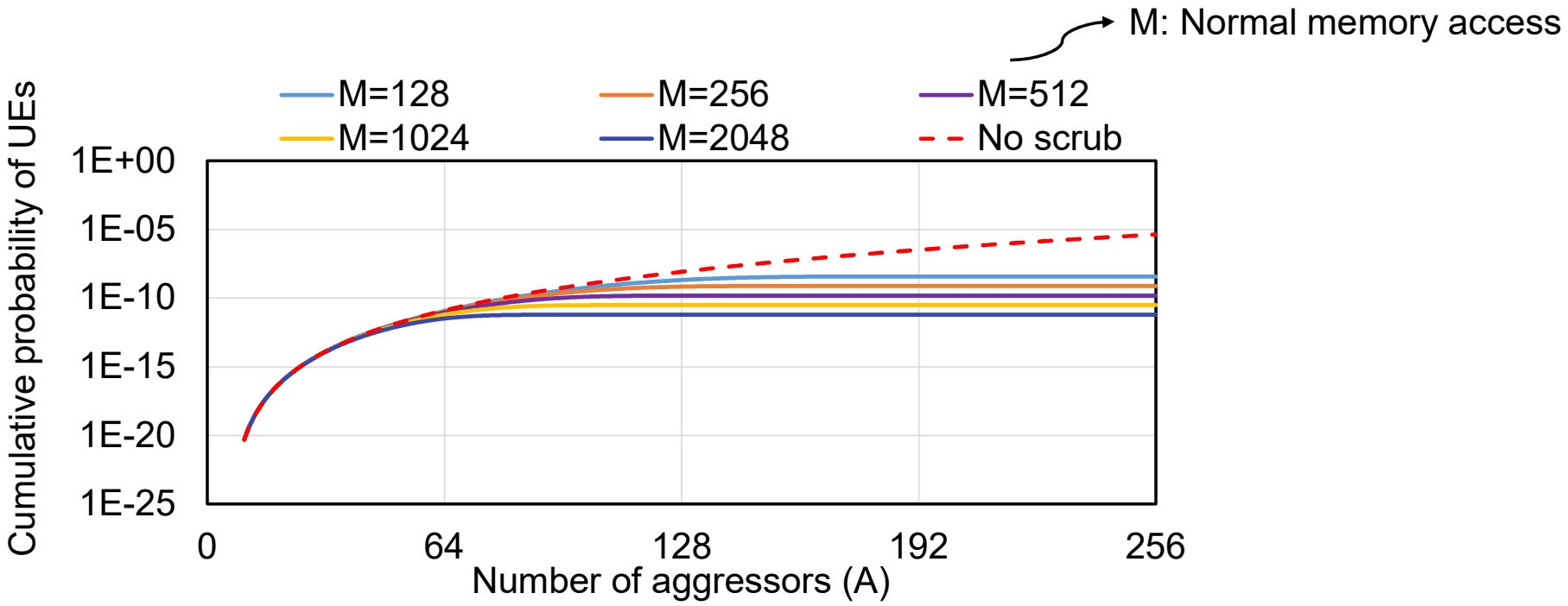
- Guardband-based **demand scrubbing**
- Instead of periodic scrubbing (Cube), RowArmor uses OOC ECC corrections to estimate error accumulation.

Guardband scrubbing

- Effectiveness of **guardband scrubbing** under an adversarial access pattern

Guardband scrubbing

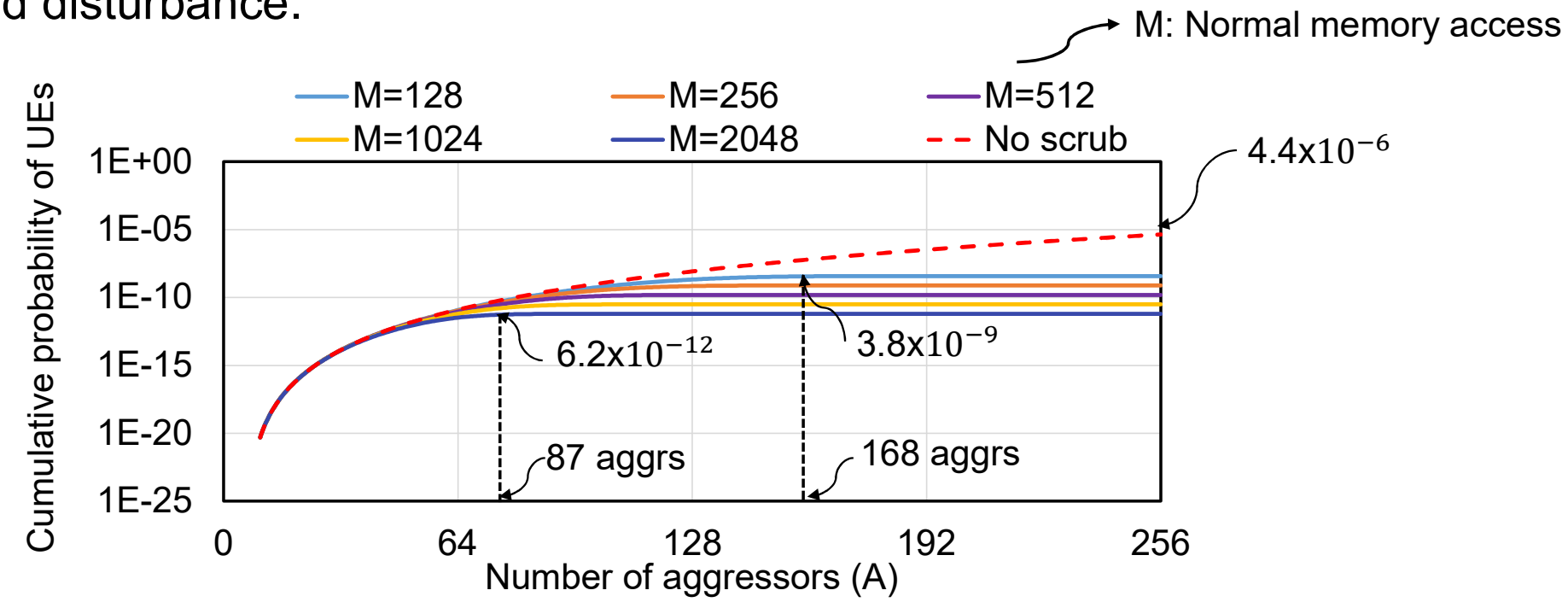
- Effectiveness of **guardband scrubbing** under an adversarial access pattern



**Uncorrectable Errors (UEs)*

Guardband scrubbing

- Effectiveness of **guardband scrubbing** under an adversarial access pattern
- Guardband scrubbing effectively halts cumulative error growth, even under sustained disturbance.



*Uncorrectable Errors (UEs)

Guardband scrubbing

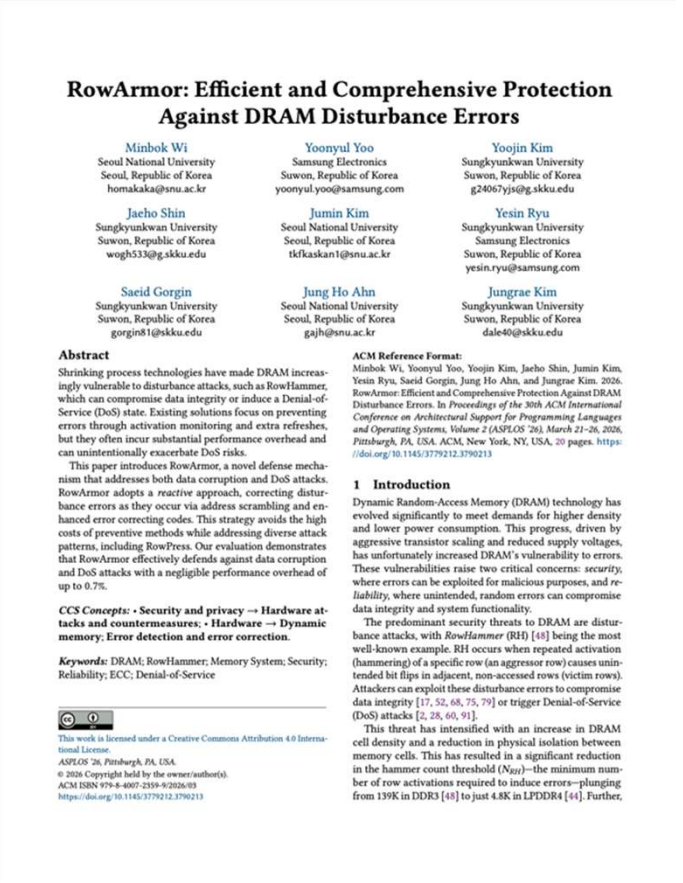
- Scrubbing a single bank may take approximately **0.1s (expensive)**.

Guardband scrubbing

- Scrubbing a single bank may take approximately **0.1s (expensive)**.
  - Solution 1) throttling suspicious threads
  - Solution 2) per-bank obfuscation key rotation

More details in the paper

- Octet scrambling
  - DAS coefficient constraints
  - SWLP permutations equations
- RAO
  - RAO design
  - Detailed effectiveness analysis
- Guardband scrubbing
  - Effectiveness under adversarial pattern
  - Suspicious threads throttling
  - Per-bank obfuscation key rotation



RowArmor: Efficient and Comprehensive Protection Against DRAM Disturbance Errors

Minbok Wi
Seoul National University
Seoul, Republic of Korea
homakaka@snu.ac.kr

Yoonyul Yoo
Samsung Electronics
Suwon, Republic of Korea
yoonyul.yoo@samsung.com

Yoojin Kim
Sungkyunkwan University
Suwon, Republic of Korea
g24067yjs@g.skku.edu

Jaeho Shin
Sungkyunkwan University
Suwon, Republic of Korea
wogh533@g.skku.edu

Jumin Kim
Seoul National University
Seoul, Republic of Korea
tkfkaskan1@snu.ac.kr

Yesin Ryu
Sungkyunkwan University
Samsung Electronics
Suwon, Republic of Korea
yesin.ryu@samsung.com

Saeid Gorgin
Sungkyunkwan University
Suwon, Republic of Korea
gorgin81@skku.edu

Jung Ho Ahn
Seoul National University
Seoul, Republic of Korea
gajh@snu.ac.kr

Jungrae Kim
Sungkyunkwan University
Suwon, Republic of Korea
dale40@skku.edu

Abstract

Shrinking process technologies have made DRAM increasingly vulnerable to disturbance attacks, such as RowHammer, which can compromise data integrity or induce a Denial-of-Service (DoS) state. Existing solutions focus on preventing errors through activation monitoring and extra refreshes, but they often incur substantial performance overhead and can unintentionally exacerbate DoS risks.

This paper introduces RowArmor, a novel defense mechanism that addresses both data corruption and DoS attacks. RowArmor adopts a *reactive* approach, correcting disturbance errors as they occur via address scrambling and enhanced error correcting codes. This strategy avoids the high costs of preventive methods while addressing diverse attack patterns, including RowPress. Our evaluation demonstrates that RowArmor effectively defends against data corruption and DoS attacks with a negligible performance overhead of up to 0.7%.

CCS Concepts: • **Security and privacy → Hardware attacks and countermeasures**; • **Hardware → Dynamic memory**; **Error detection and error correction**.

Keywords: DRAM; RowHammer; Memory System; Security; Reliability; ECC; Denial-of-Service

ASPLOS '26, Pittsburgh, PA, USA.

ACM ISBN 979-8-4007-2359-9/2026/03
https://doi.org/10.1145/3779212.3790213

ACM Reference Format:
Minbok Wi, Yoonyul Yoo, Yoojin Kim, Jaeho Shin, Jumin Kim, Yesin Ryu, Saeid Gorgin, Jung Ho Ahn, and Jungrae Kim. 2026. RowArmor: Efficient and Comprehensive Protection Against DRAM Disturbance Errors. In *Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '26), March 21–26, 2026, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3779212.3790213

1 Introduction

Dynamic Random-Access Memory (DRAM) technology has evolved significantly to meet demands for higher density and lower power consumption. This progress, driven by aggressive transistor scaling and reduced supply voltages, has unfortunately increased DRAM's vulnerability to errors. These vulnerabilities raise two critical concerns: *security*, where errors can be exploited for malicious purposes, and *reliability*, where unintended, random errors can compromise data integrity and system functionality.

The predominant security threats to DRAM are disturbance attacks, with *RowHammer* (RH) [48] being the most well-known example. RH occurs when repeated activation (hammering) of a specific row (an aggressor row) causes unintended bit flips in adjacent, non-accessed rows (victim rows). Attackers can exploit these disturbance errors to compromise data integrity [17, 52, 68, 75, 79] or trigger Denial-of-Service (DoS) attacks [2, 28, 60, 91].

This threat has intensified with an increase in DRAM cell density and a reduction in physical isolation between memory cells. This has resulted in a significant reduction in the hammer count threshold (N_{RH})—the minimum number of row activations required to induce errors—plunging from 139K in DDR3 [48] to just 4.8K in LPDDR4 [44]. Further,

Recap

- RowArmor integrates four mechanisms:
  1. Octet scrambling (DAS & SWLP)
  2. Octuple-Octet Correcting ECC (OOC ECC)
  3. Row Address Obfuscation (RAS)
  4. Guardband scrubbing

RowArmor: Evaluation

Evaluation methodology

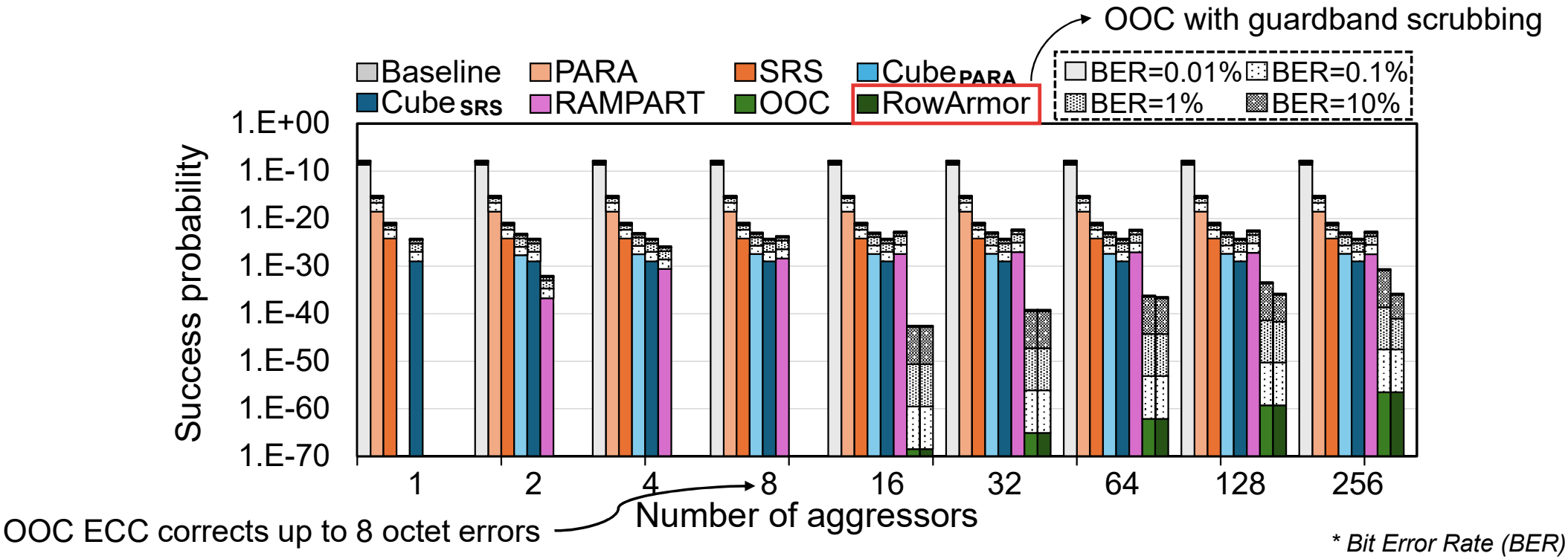
- **Security**: Targeted and Denial of Service (DoS) RH attacks
- **Performance**: Slowdown under benign workloads
- **Reliability**: Protection against random DRAM errors
- **Hardware overhead**: MC and DRAM area cost

Security

- Probability of successful **targeted attacks**.
  - 24-hour attacks with different Bit Error Rate (BER).
  - An attack succeeds if ECC neither corrects nor detects the induced errors.

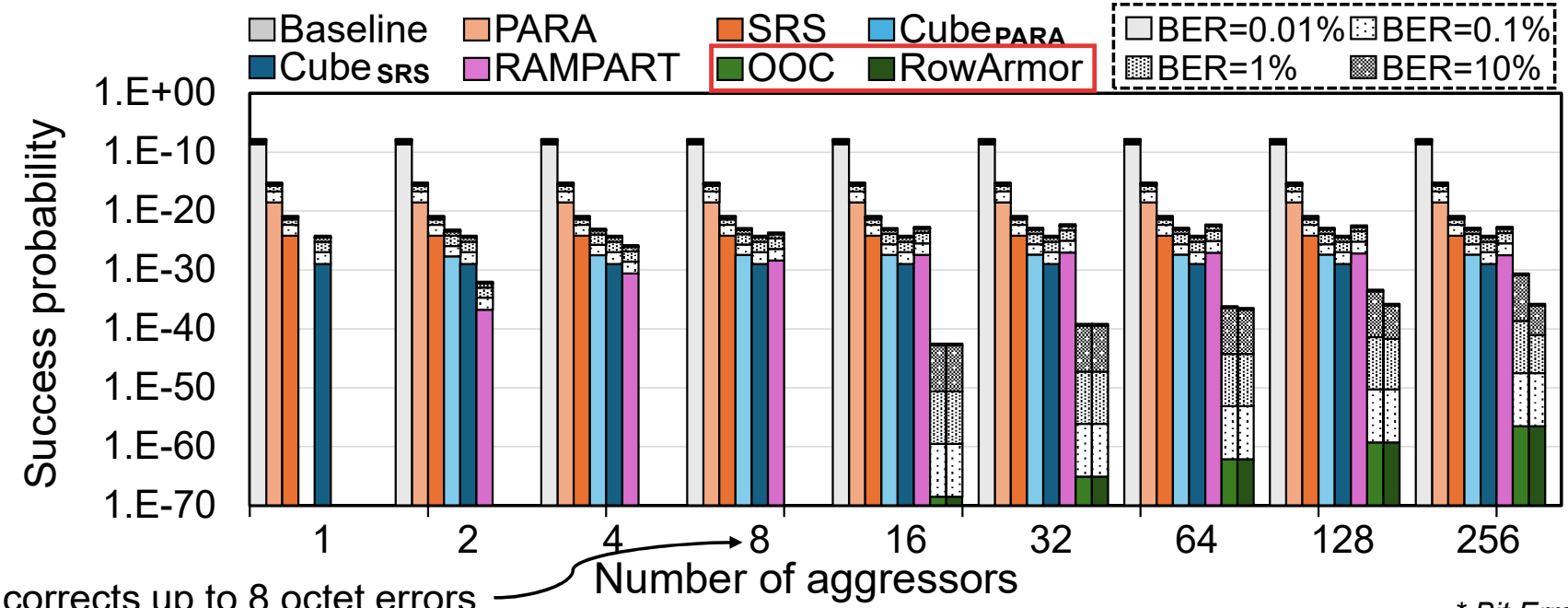
Security

- OOC ECC guarantees integrity **up to 8 aggressors**.
- RowArmor consistently achieves the **lowest** attack success probability.



Security

- Limited impact of guardband scrubbing at low BER (0.01%) due to fewer observable errors.

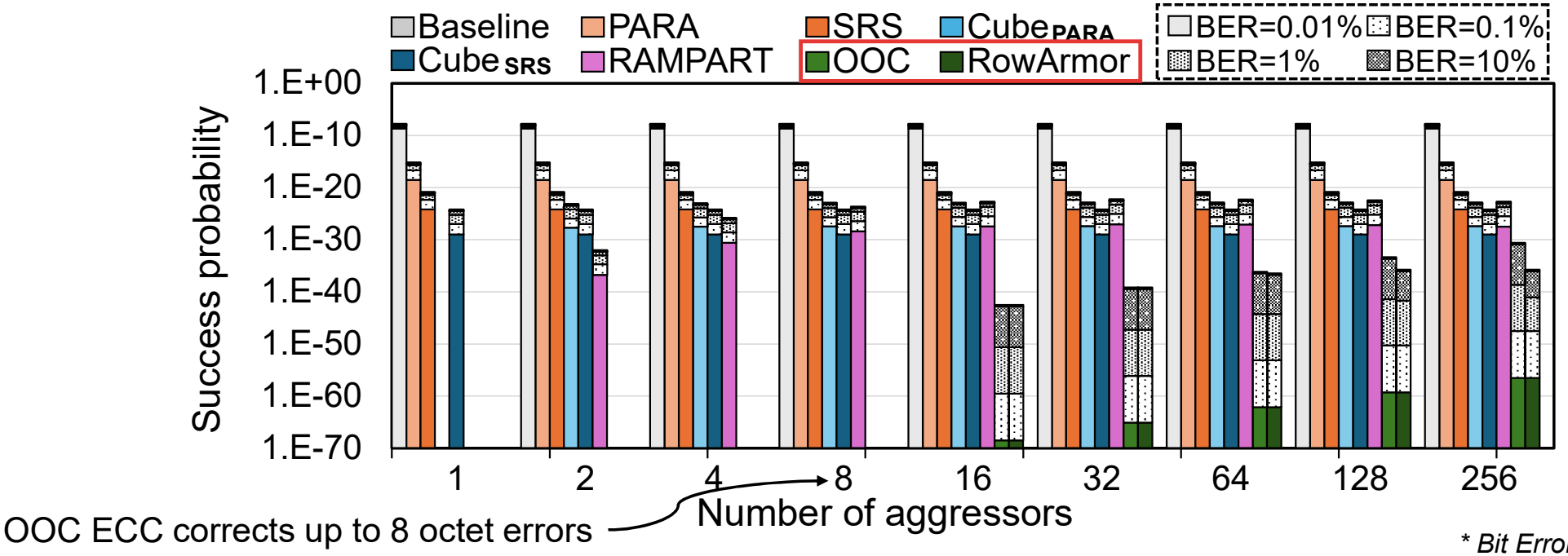


OOC ECC corrects up to 8 octet errors

** Bit Error Rate (BER)*

Security

- Higher BER challenges RowArmor, as most octets contain bit errors.
  - Guardband scrubbing becomes more effective at high BER.



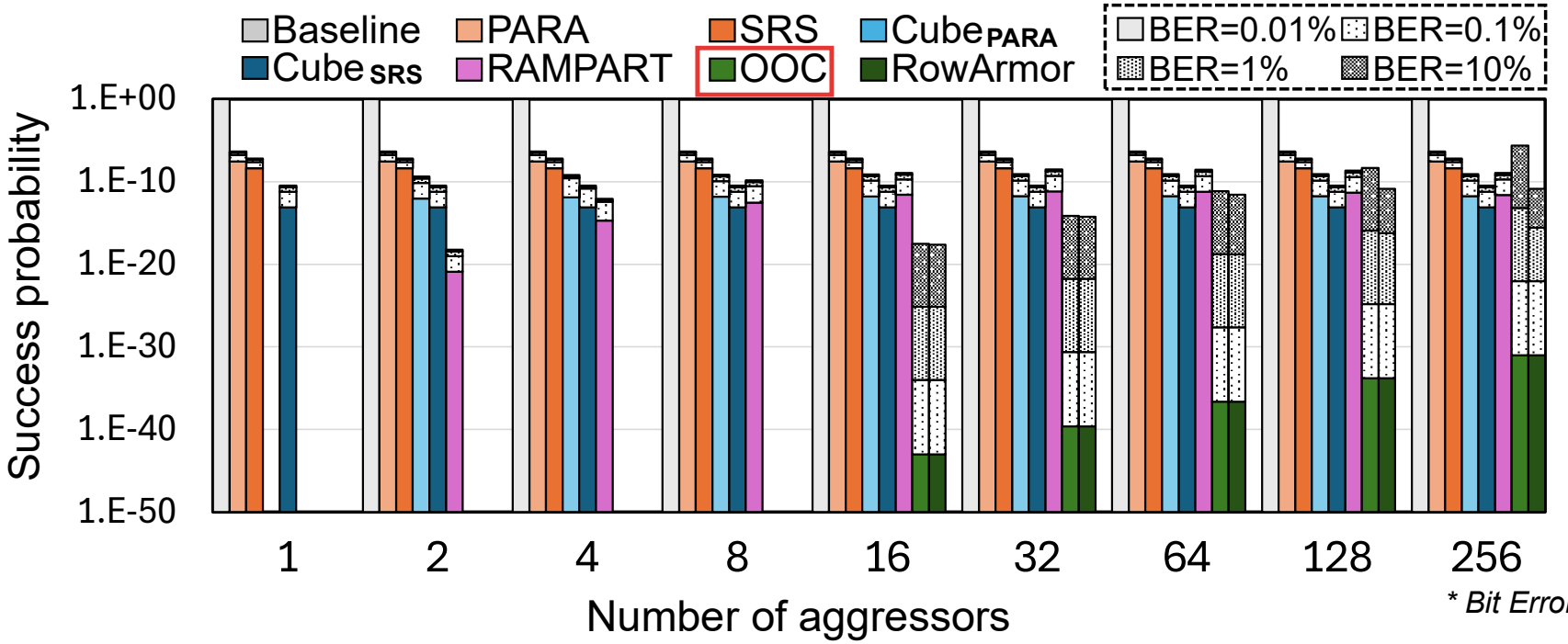
** Bit Error Rate (BER)*

Security

- Probability of successful **DoS attacks**.
  - An attack succeeds either generating Detected but Uncorrectable Errors *(*DUEs) or triggering excessive preventive actions.

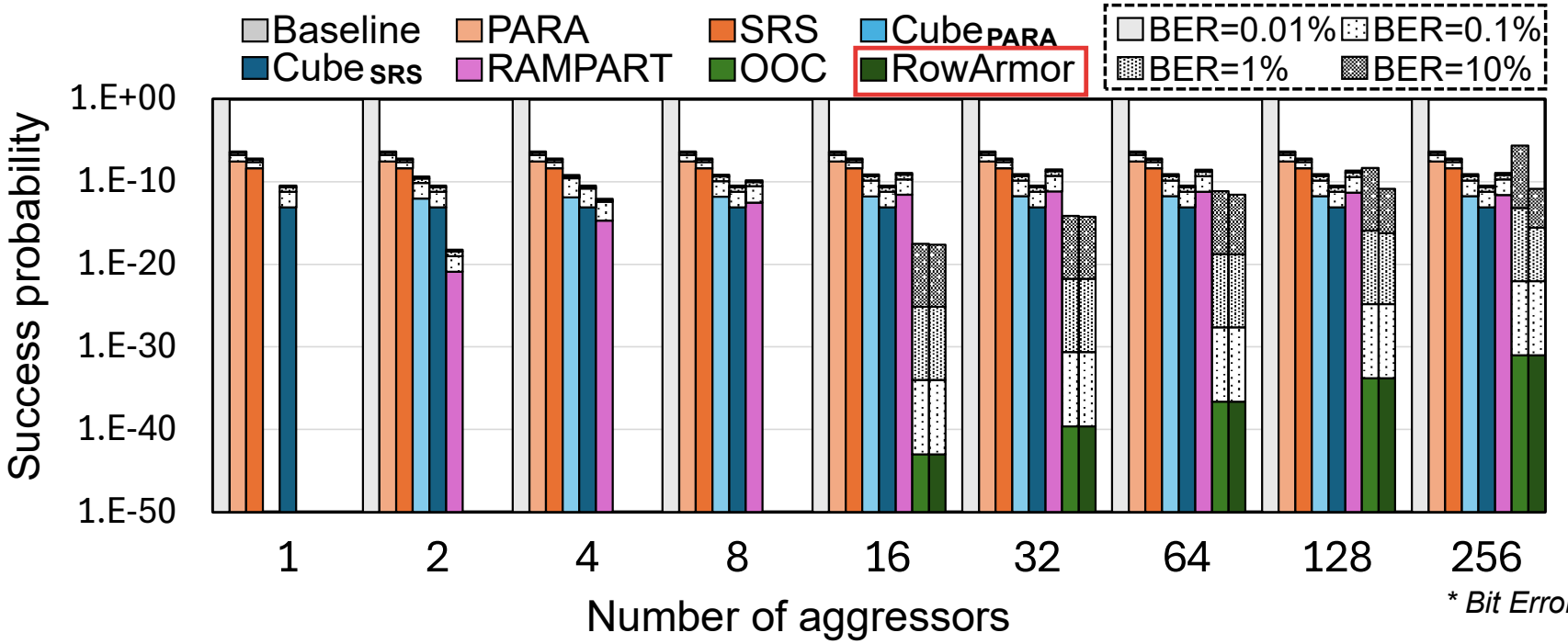
Security

- Probability of successful **DoS attacks**.
  - OOC only the success probability becomes higher than that of some prior schemes.



Security

- Probability of successful **DoS attacks**.
  - OOC only the success probability becomes higher than that of some prior schemes.
  - RowArmor achieves **lowest** success probability of DoS attacks.



** Bit Error Rate (BER)*

System performance

- Simulator: McSimA+ [1]
- Metric: Normalized weighted speedup
- Workloads: SPEC CPU2017 [2] trace and categorized into Rate, Mix-High, Mix-Random.

Parameter	Value
Processor	16 OoO cores @3.2GHz, 4-way issue
LLC	16MB / 16-way / shared
Main Memory	DDR5-6400C (CL-nRCD-nRP: 56-56-56) 32GB / 2 channels / 2 ranks per channel

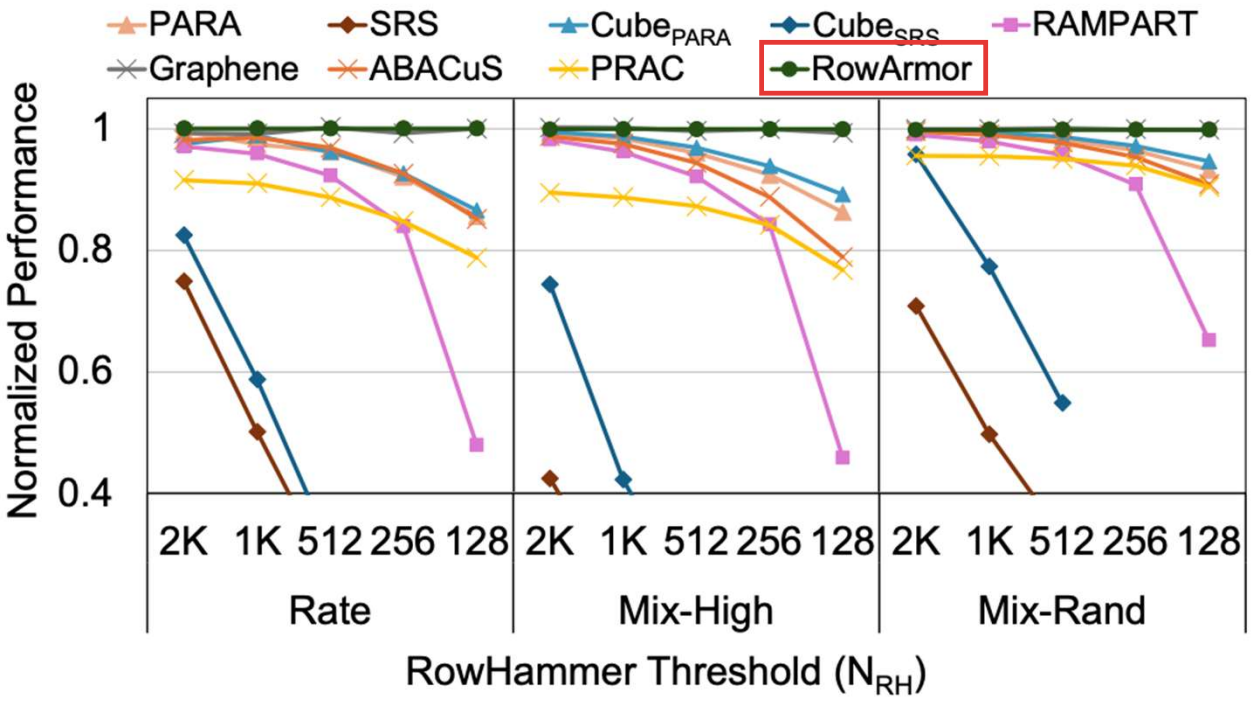
[1] Jung Ho Ahn, Sheng Li, and Norman P. Jouppi. "McSimA+: A manycore simulator with application-level+ simulation and detailed microarchitecture modeling." *ISPASS*. IEEE, 2013.
[2] Bucek, James, Klaus-Dieter Lange, and Jóakim v. Kistowski. "SPEC CPU2017: Next-generation compute benchmark." *Companion of the 2018 ACM/SPEC International Conference on Performance Engineering*. 2018.

System performance

- RowArmor increases t_{RCD} and read latency by **only one cycle** each.

System performance

- RowArmor achieves minimal performance overhead (≈ **0.7%** for Mix-High).



Reliability

- Metric:
  - Correcting Error (**CE**) / Detected but Uncorrectable Error (**DUE**) / Silent Data Corruption (**SDC**) rates via Monte-Carlo simulation.
- Compared ECC schemes:
  - AMD Chipkill, Bamboo QPC, RowArmor
- Error types evaluated:
  - Bit errors, Chip errors, Rank errors

Reliability

- All schemes correct single bit/pin/chip errors.

(72,64) (80,64)

Error scenario	Decoding result	AMD Chipkill	Bamboo QPC	RowArmor
1 bit/1 pin/1 chip	CE (%)	**100.0000**	**100.0000**	**100.0000**
2 × 1 bit	CE (%)	9.8858	**100.0000**	**100.0000**
	DUE (%)	89.6274	0.0000	0.0000
	SDC (%)	0.4868	0.0000	0.0000
3 × 1 bit	CE (%)	0.9586	76.7570	**100.0000**
	DUE (%)	97.6827	23.2430	0.0000
	SDC (%)	1.3587	0.0000	0.0000
1 chip + 1 bit	CE (%)	0.0000	10.0017	12.4164
	DUE (%)	96.8736	89.9983	87.5836
	SDC (%)	3.1264	0.0000	0.0000
1 chip + 2 bits	CE (%)	0.0000	0.9982	1.5882
	DUE (%)	99.3873	99.0018	98.4118
	SDC (%)	0.6127	0.0000	0.0000
1 chip + 1 chip	DUE (%)	**100.0000**	**100.0000**	**100.0000**
rank	DUE (%)	**100.0000**	**100.0000**	**100.0000**

Reliability

- All schemes correct single bit/pin/chip errors.
- Compared to Bamboo QPC, OOC provides stronger protection with its **larger codewords**.

(72,64) (80,64)

Error scenario	Decoding result	AMD Chipkill	Bamboo QPC	RowArmor
1 bit/1 pin/1 chip	CE (%)	**100.0000**	**100.0000**	**100.0000**
2 × 1 bit	CE (%)	9.8858	**100.0000**	**100.0000**
	DUE (%)	89.6274	0.0000	0.0000
	SDC (%)	0.4868	0.0000	0.0000
3 × 1 bit	CE (%)	0.9586	76.7570	**100.0000**
	DUE (%)	97.6827	23.2430	0.0000
	SDC (%)	1.3587	0.0000	0.0000
1 chip + 1 bit	CE (%)	0.0000	10.0017	12.4164
	DUE (%)	96.8736	89.9983	87.5836
	SDC (%)	3.1264	0.0000	0.0000
1 chip + 2 bits	CE (%)	0.0000	0.9982	1.5882
	DUE (%)	99.3873	99.0018	98.4118
	SDC (%)	0.6127	0.0000	0.0000
1 chip + 1 chip	DUE (%)	**100.0000**	**100.0000**	**100.0000**
rank	DUE (%)	**100.0000**	**100.0000**	**100.0000**

Hardware overhead

- RTL implementation of **Octet scrambling (in DRAM) and RAO (in MC)**
  – Synthesizes using Synopsys design compiler (UMC 28 nm)
- DRAM overhead: less than **0.005%** die area (vs. PRAC ≈ 0.1%)
- MC overhead: less than **0.00001%** die area (vs. Hydra ≈ 0.005%)

Summary

- RowArmor: Stronger ECC enables RowHammer protection **without preventive mitigation**.
  - Octet Scrambling
  - OOC ECC
  - Row Address Obfuscation
  - Guardband Scrubbing
- RowArmor provides strong disturbance protection with **minimal** performance and hardware cost.

Thank you!

Backup Slide

Comparison with Cube [1]

	Cube [1]	RowArmor
ECC	OECC + Chipkill	OECC + **OOC ECC**
Address scrambling	Chip-wise scrambling	**Octet-wise** (8bit) scrambling
Obfuscation	Feistel cipher (MC)	Feistel cipher (MC)
Scrubbing	Periodic OECC scrubbing	Triggered by OOC ECC corrections (**Guardband scrubbing**)
ECC against Multi-aggressor attack	Single aggressor attack	Secure **up to 8-aggressor** attack
Need preventive mitigation	Yes (PARA or SRS)	**No**

[1] Kim, Michael Jaemin, et al. "How to kill the second bird with one ecc: The pursuit of row hammer resilient dram." *MICRO*. 2023.

Example of octet scrambling

Chip	DQ	Octet	Row addr. scramble	SWL permutation	Overall scramble
0	0	0	$DA \times 1$	$DA \times 1$	$DA \times 1$
		1		$DA \times 3$	$DA \times 3$
	1	2	$DA \times 5$	$DA \times 1$	$DA \times 5$
		3		$DA \times 3$	$DA \times 15$
	2	4	$DA \times 9$	$DA \times 1$	$DA \times 9$
		5		$DA \times 3$	$DA \times 27$
	3	6	$DA \times 13$	$DA \times 1$	$DA \times 13$
		7		$DA \times 3$	$DA \times 39$
1	4	8	$DA \times 17$	$DA \times 1$	$DA \times 17$
		9		$DA \times 3$	$DA \times 51$
⋮	⋮	⋮	⋮	⋮	⋮

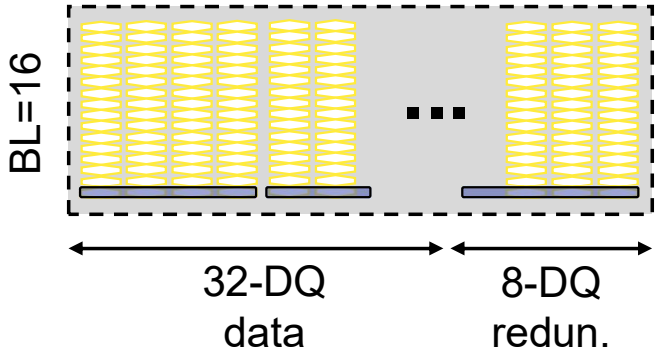
Octet scrambling considering Blast Radius (BR)

- Needs more constraints on octet scrambling coefficient.
- Considering subarray.

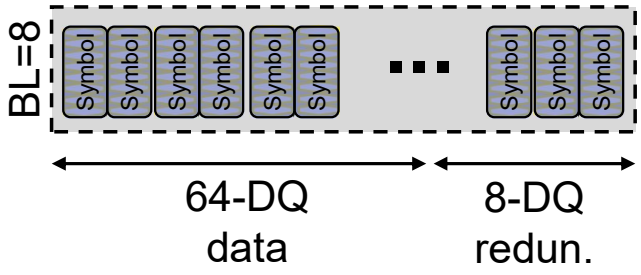
ECC layout

- Chipkill
  - Chip-level protection
  - Corrects 1 chip failure
- Bamboo ECC
  - DQ level protection
  - Corrects 1 DQ failure
- **OOC ECC** (RowArmor)
  - Octet-level symbols
  - Matches scrambled data layout
  - Corrects up to 8 octet errors

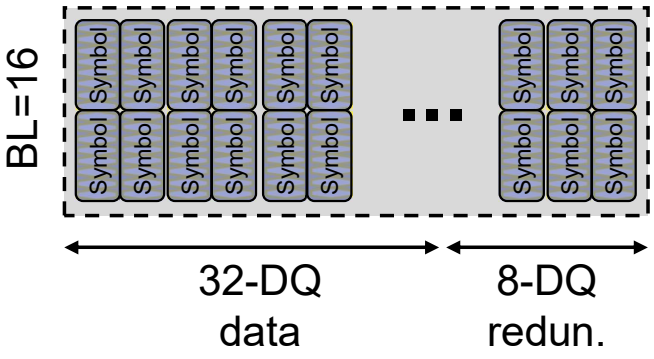
An ECC codeword with
(8+2) symbols



An ECC codeword with
(64+8) symbols

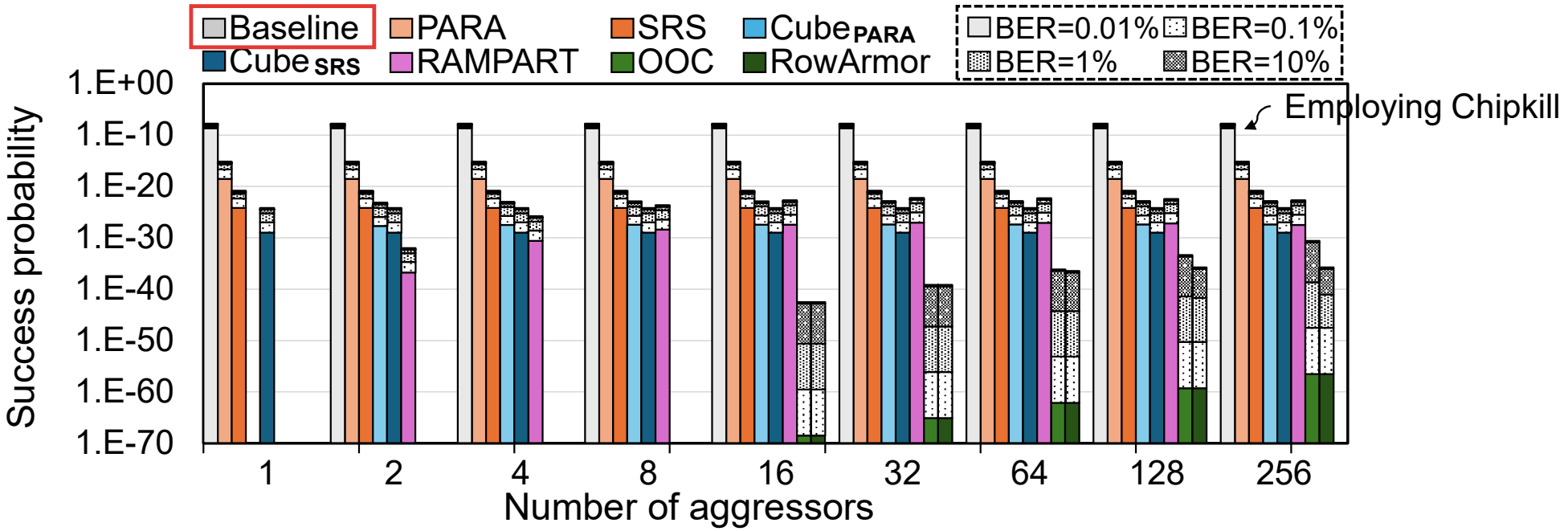


An ECC codeword with
(64+16) symbols



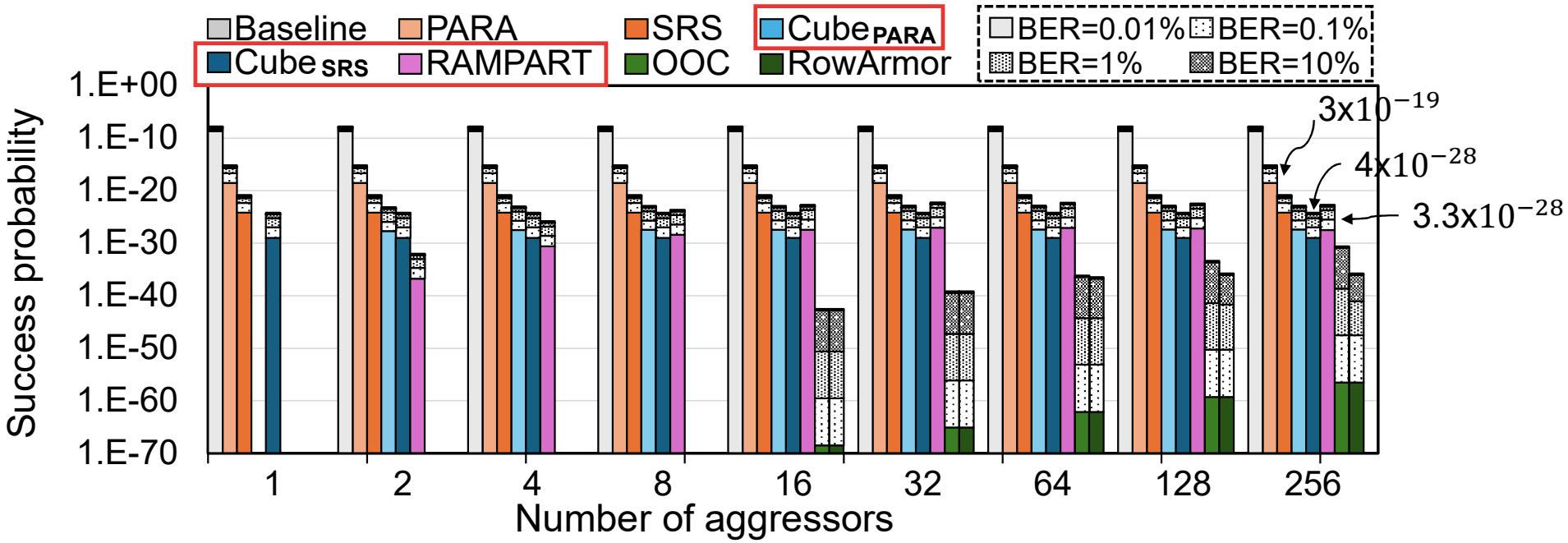
Security

- The baseline (no protection), employ a modified AMD Chipkill ECC.
  - Achieves $2.3\text{x}10^{-9}$ success probability



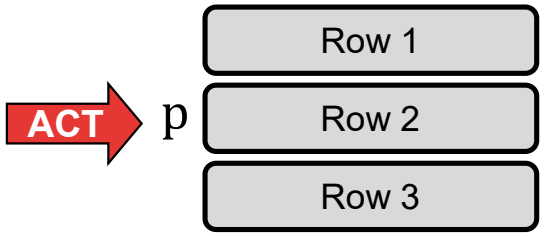
Security

- Address obfuscation prevents precise targeting.
  - Cube and Rampart offer stronger protection than PARA and SRS.



Preventive mitigation schemes

- (1) Preventively detect potential RH threat
  - Keep row activations (ACT count) threshold below N_{RH} to trigger mitigation action

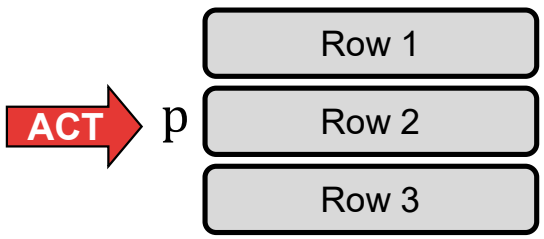


Probability of triggering Mitigation action (p)[1]

[1] Y. Kim et al., "Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors," ISCA, 2014.

Preventive mitigation schemes

- (1) Preventively detect potential RH threat
  - Keep row activations (ACT count) threshold below N_{RH} to trigger mitigation action



Probability of triggering Mitigation action (p)[1]

Row	# of ACTs
0x4	10
0xc	4

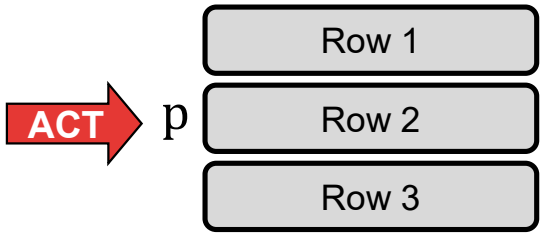
* threshold = 10

Counter Table (In DRAM or MC) [2]

[1] Y. Kim et al., "Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors," ISCA, 2014.
[2] JEDEC, "JESD79-5C.01_v1.31," 2024.

Preventive mitigation schemes

- (1) Preventively detect potential RH threat
  - Keep row activations (ACT count) threshold below N_{RH} to trigger mitigation action

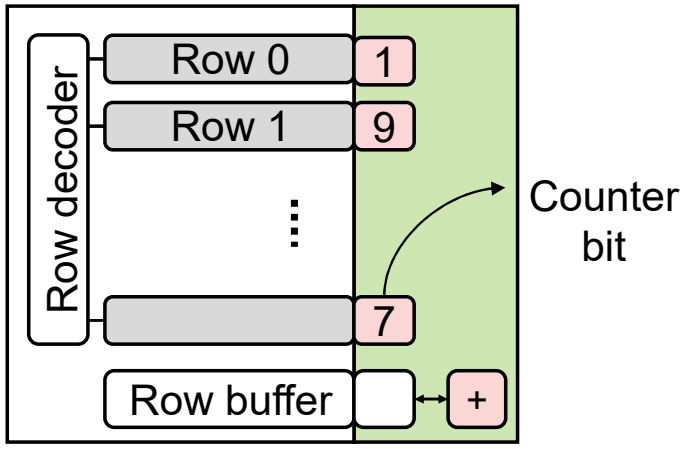


Probability of triggering Mitigation action (p)[1]

Row	# of ACTs
0x4	10
0xc	4

* threshold = 10

Counter Table (In DRAM or MC) [2]



Per-Row Activation Counting (PRAC) [2]

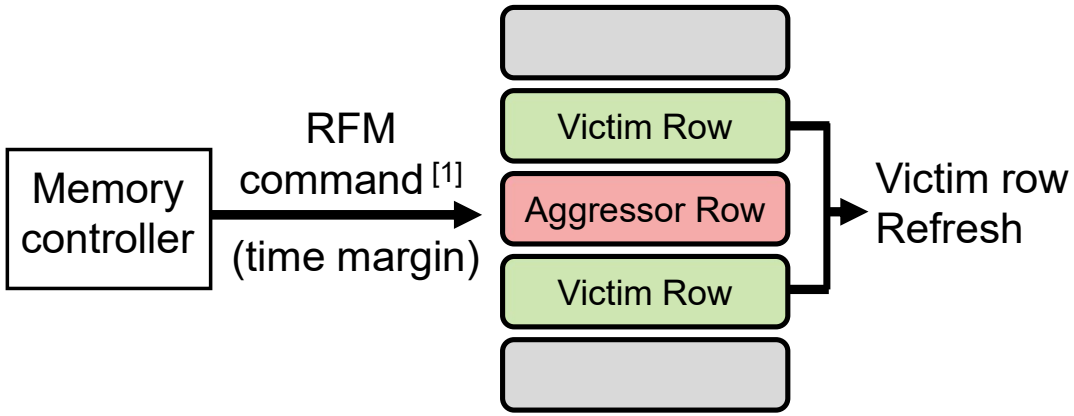
[1] Y. Kim et al., "Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors," ISCA, 2014.
[2] JEDEC, "JESD79-5C.01_v1.31," 2024.

Preventive mitigation schemes

- (2) Trigger mitigation actions
  - Issue mitigation actions

Preventive mitigation schemes

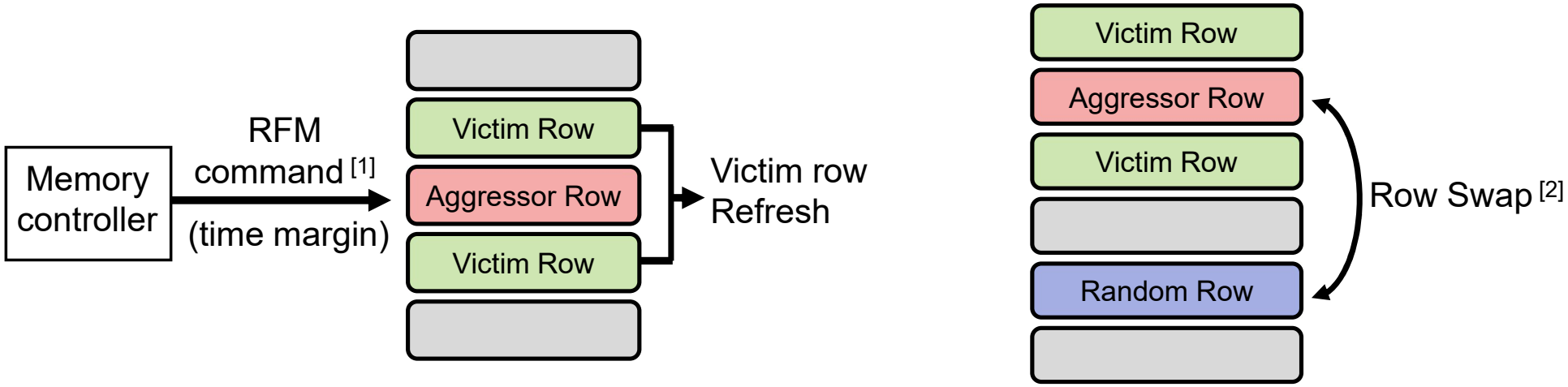
- (2) Trigger mitigation actions
  - Victim row refresh



[1] JEDEC, "JESD79-5C.01_v1.31," 2024.

Preventive mitigation schemes

- (2) Trigger mitigation actions
  - Victim row refresh, row swap



[1] JEDEC, "JESD79-5C.01_v1.31," 2024.
[2] Saileshwar, Gururaj, et al. "Randomized row-swap: mitigating row hammer by breaking spatial correlation between aggressor and victim rows." *ASPLOS*. 2022.

Correction Validation for Reliability

- OOC ECC maximizes correction capability to tolerate multi-aggressor attacks.
- However, aggressive correction may increase the risk of *miscorrection* and potential *silent data corruption* from random faults.
- RowArmor introduces **Correction Validation**:
  - Detect unlikely error patterns.
  - Check DRAM OD-ECC error counters.
  - Distinguish disturbance errors from random faults.
- Balancing security and reliability.